NIME 2011

Proceedings of the International Conference on New Inferfaces for Musical Expression

NIME 2011 OSLO

Proceedings of the International Conference on New Interfaces for Musical Expression

30 May – 1 June 2011
Oslo, Norway

Edited by:
Alexander Refsum Jensenius
Anders Tveit
Rolf Inge Godøy
Dan Overholt

Proceedings published by
Department of Musicology, University of Oslo
Norwegian Academy of Music

Copies may be ordered from:
Department of Musicology
P.O. Box 1017 Blindern
N-0315 Oslo, Norway

Web sites
www.nime2011.org
www.nime.org

Cover design
Thomas Kjellberg

ISSN 2220-4792 (Print)
ISSN 2220-4806 (Online)
ISSN 2220-4814 (USB)

UiO : University of Oslo

NORGESMUSIKKHØGSKOLE
Norwegian Academy of Music

BEK

notam.

NTNU – Trondheim
Norwegian University of
Science and Technology

[simula . research laboratory]

norsk teknisk museum

cost
EUROPEAN COOPERATION
IN SCIENCE AND TECHNOLOGY

The Research Council
of Norway

NORSK KULTURRÅD
Arts Council Norway

I-CUBE X

NORWEGIAN
FILM INSTITUTE
NFI.no

cowave

NaturalPoint®

QUALISYS
Motion Capture Systems

Welcome to NIME

On behalf of the University of Oslo, the Norwegian Academy of Music and our partners and sponsors, we are proud to present the 11th edition of NIME.

In its 11th year NIME has become an important conference series, the meeting point of researchers, developers and artists from all over the world. Even though participants come from widely different backgrounds, they share a mutual interest in groundbreaking music and technology.

Since the start of the conference series, the word *NIME* has started to take on meaning in itself, independent of the annual conferences. Even outside the core group of annual conference participants, people start to know that NIME is somehow related to exciting musical and artistic research and practice. Still, though, *NIME* is mainly used as a noun, e.g. "bring your favourite NIMEs to the jam session tonight." Perhaps it is now time to start using it also as a verb: *to nime*.

What's in a word? We are often asked by people what NIME actually means. There is an official answer, but we rather like the idea that the four letter acronym can take on new meanings:

- N = New, Novel, ...
- I = Interfaces, Instruments, ...
- M = Musical, Multimedial, ...
- E = Expression, Exploration, ...

Whatever the meaning of the letters, the underlying idea is the hunt for new understanding, development and artistic exploration of devices in music. This type of exploration in (and on the borders between) science and art is not a problem for people in the NIME community. Outside the NIME community, however, our experience is that the worlds of science and art are often separate. We believe that the conference and the community can make a difference, and show that science and art need each other to prosper.

The NIME conference has over the last decade grown from a workshop at CHI in 2001, to have more than 500 submissions in 2011. This record number of submissions has made it possible to set up a large and varied program that we hope will be inspiring for everyone being present. Despite the large submission number, we decided to keep NIME as an "intimate" conference, a conference where it is possible to attend everything. We have stuck with the idea of single-track presentations, even though this means that the acceptance rate for oral presentation was as low as 15%. Keeping with the NIME spirit, though, we think that large poster and demo sessions are probably the best way of seeing, testing, and exploring various new instruments/interfaces in practice.

Programming concerts for a conference like NIME is challenging. Here we have tried to find a balance between novel instruments, performance maturity, and artistic expression. We were happy to see that many picked up on our challenge of submitting combined "paper + performance" proposals. These submissions were treated as two separate submissions at first, through separate scientific and artistic review, before being evaluated together. Some were accepted together, and some were accepted as either paper or performance. Keeping up to international standards on both a scientific paper and a performance is not an easy task. But we see that many people in the community are up for it, and we highly encourage this type of double submissions also for future conferences.

There will be three keynote lectures, all of which will approach the topic of NIME from different angles. Tellef Kvifte's lecture will bring in historical and organological perspectives through a discussion of digital instruments in the 19th century. David Rokeby will talk about his exploration of using the body in interactive art, something which is currently very popular in the NIME community. Sergi Jordà will talk about his instruments, and possibilities/challenges in working between science, art and industry. We hope these lectures will be inspiring and help to draw some longer lines in between the shorter and more fast-paced presentations at the conference.

We are also happy to present a series of pre-NIME events. As usual, there is a large set of tutorials and workshops held by local and international NIME participants. This year there is also a symposium called "Technology and Aesthetics," produced by NOTAM. The Art.on.Wires Media festival is a 5 day long feast of hacking, lectures and concerts, produced by the newly established Art.on.Wires socity. Finally, there is an exhibition on Sonic Interaction Design produced by COST Action IC0601 and BEK.

A number of organisations and individuals have contributed to making this conference a reality. There is only one thing to say: thank you!

All in all, we hope that NIME 2011 will represent another milestone in the development of the NIME community, and be of inspiration to all of you who participate. Happy NIMEing.

Alexander Refsum Jensenius & *Kjell Tore Innervik*
University of Oslo & Norwegian Academy of Music

Organizers

Institutions

University of Oslo
Department of Musicology

Norwegian Academy of Music

NOTAM
Norwegian Centre for Technology in Music and Arts

BEK
Bergen Center for Electronic Arts

NTNU
Norwegian University of Science and Technology

Simula Research Laboratory

Norwegian Museum of Science, Technology and Medicine

COST Action IC0601

Chairs

Conference chairs:
Alexander Refsum Jensenius
Kjell Tore Innervik

Paper chairs:
Alexander Refsum Jensenius
Rolf Inge Godøy

Music chairs:
Kjell Tore Innervik
Ivar Frounberg

Demonstration chair:
Dan Overholt

SID exhibition curators:
Trond Lossius
Frauke Behrendt

Symposium chairs:
Notto J. W. Thelle
Jøran Rudi
Rune Molvær

Art.on.Wires chair:
Alexander Eichhorn

Local Committee

Frauke Behrendt
Øyvind Brandtsegg
Alexander Eichhorn
Ivar Frounberg
Rolf Inge Godøy
Trond Lossius
Dan Overholt
Jøran Rudi
Jim Tørresen

Steering Committee

Frédéric Bevilacqua
Tina Blaine
Sidney Fels
Michael Lyons
Sile O'Modhrain
Yoichi Nagashima
Joe Paradiso
Carol Parkinson
Norbert Schnell
Eric Singer
Atau Tanaka

Local organization

Arjun Chandra
Anette Pauline Forsbakk
Kyrre Glette
Yngve Hafting
Mats Høvin
Thomas Kjellberg
Cato Langnes
Kristian Nymoen
Otto Christian Pay
Ståle A. Skogstad
Renate Hauge Sund
Siren Tjøtta
Anders Tveit
Knut Vik
Arve Voldsund
Anne Cathrine Wesnes
Ellen Wingerei
Alison Bullock Aarsten

Symposium

Asbjørn Blokkum Flø
Cato Langnes
Rune Molvær
Jøran Rudi
Henrik Sundt
Notto J. W. Thelle
Hans Wilmers

Art.on.Wires

Ulli Dibowski
Alexander Eichhorn
Jason Geistweidt

SID exhibition

Dag Andreassen
Daniel Arfib
Maria Grazia Ballerano
Frauke Behrendt
Anne Marthe Dyvi
Espen Egeland
Elisabeth Gmeiner
Thomas Hermann
Trond Lossius
Monique Mossefinn
Alessandra Paccamiccio
Inge de Prins
Matteo Razzanelli
Davide Rocchesso
Aranzazu Sanchez
Henning Sandsdalen
Lars Ove Toft
Marieke Verbiesen
Frode Weium

All names are in alphabetical order.

NIME reviewers

Sarah Fdili Alaoui
Jesse Allison
Anders Andersson

Frauke Behrendt
Ross Bencina
Edgar Berdahl
Andreas Bergsland
Eirik Birkeland
Tina Blaine
Ben Bogart
Sinan Bokesoy
Bert Bongers
Brennon Bortz
Mathieu Bosi
Nicolas Bouillot
Øyvind Brandtsegg
Roberto Bresin
Nick Bryan-Kinns
Gaspard Bucher
Eivind Buene
Ivica Bukvic
Jamie Bullock
Sinan Bökesoy
Niels Böttcher

Baptiste Caramiaux
Alvaro Cassinelli
Parag Chordia
Mats Claesson
Michael Cohen
Graham Coleman
Nick Collins
Langdon Crawford
Alain Crevoisier

Nicolas D'Alessandro
Palle Dahlstedt
Roger Dannenberg
Scott Deal
Smilen Dimitrov
Paul Doornbusch
Luke Dubois

Alexander Eichhorn
Trond Engum
Georg Essl

Sidney Fels
Robin Fencott
Rebecca Fiebrink
Wolfgang Fohl
Federico Fontana
Angelo Fraietta
Alexandre Francois
Adrian Freed
Jason Freeman
Anders Friberg
Henrik Frisk
Ivar Frounberg
Ichiro Fujinaga
Andrew Cavan Fyans
Chris Geiger
Steven Gelineck
David Gerhard
Kyrre Glette
Rolf Inge Godøy
Lars Graugaard
Tobias Grosshauser
Sylvain le Groux
Carlos Guedes
Michael Gurevich

Bjørnar Habbestad
Aristotelis Hadjakos
Morten Halle
Tor Halmrast
Keith Hamel
Kjetil Falkenberg Hansen
Mark Havryliv
Andrew Hawryshkewich
Tomás Henriques
Saburo Hirano
Hannes Hoelzl
Risto Holopainen
Mark David Hosale
Bill Hsu
Mats Høvin

Kjell Tore Innervik

Javier Jaimovich
Jordi Janer
Alexander Refsum Jensenius
Andrew Johnston
Sergi Jorda

Haruhiro Katayose
Peter Kirn
Benjamin Knapp
Juraj Kojs
Mariusz Kozak
Tellef Kvifte

Johnathan F. Lee
Paul Lehrman
George E. Lewis
Takuro Mizuta Lippit
Trond Lossius
Michael Lyons

John Maccallum
Matthieu Macret
Thor Magnusson
Joseph Malloch
Adnan Marquez-Borbon
Mark Marshall
Kjetil Svalastog Matheussen
James Maxwell
Eduardo Miranda
Thomas B. Moeslund
Katherine Moriwaki
Florian Floyd Mueller

Yoichi Nagashima
Luiz Naveda
Kia Ng
Per Anders Nilsson
Kristian Nymoen
Sile O'Modhrain
Jieun Oh
Dan Overholt

Jyri Pakarinen
Brett Park
Philippe Pasquier
Jean-Marc Pelletier
Nils Peters
Toiviainen Petri
Timothy Place
Patrick Pogscheba

Anthony Rowe
Robert Rowe
Jøran Rudi
Even Ruud
Joel Ryan

Jan Schacher
Margaret Schedel
Norbert Schnell
Erwin Schoonderwaldt
Diemo Schwarz
Richard Scott
Stefania Serafin
Greg Shear
Stephen Sinclair
Ståle A. Skogstad
Scott Smallwood
Stefan Smulovitz
Hugo Solis
Jorge Solis
Andrew Sorensen

Hans Tammen
Peter Tornquist
Giuseppe Torre
Dan Trueman
George Tzanetakis
Jim Tørresen

Owen Vallis
Giovanna Varni
Bill Verplank
Anders Vinjar
Gualtiero Volpe

Carl Haakon Waadeland
Marcelo M. Wanderley
Ge Wang
Rob Waring
Hans Wilmers
Lonce Wyse
Björn Wöldecke

Anna Xambo

Matthew Yee-King
Tomoko Yonezawa
Takegawa Yoshinari

Mark Zadel
Michael Zbyszynski

Tone Åse

NIME selection

The NIME call for participation was published 1 September 2010, and is republished below. A total of 502 submissions were made in all categories of the call.

All submissions in the paper and performance tracks were subject to a single-blind peer review process by a group of international experts. Submissions in the performance + paper track were at first evaluated as two separate submissions, and re-evaluated as a combined submission in the final selection process.

Submissions for the SID exhibition, installations and tutorials were selected by groups of curators.

Paper track
Of 204 submissions in the different paper categories, the following have been selected:

- Oral presentation: 33
- Poster presentation: 80
- Demonstrations: 16

Other tracks
The following numbers of submissions have been selected from the other tracks (submission number in parentheses):

- Concerts: 35 (134)
- Installations 3 (33)
- Tutorials 19 (26)
- SID exhibition: 12 (102)

Call for participation

We invite you to be part of the International Conference on New Interfaces for Musical Expression. The core purpose of the NIME conference is to present the latest results in design, development, performance and analysis of/for/with new interfaces and instruments for musical use. In 2011 we will put an extra emphasis on performance aspects related to NIME, something which will also be addressed in a symposium, workshops and master classes in the days leading up to the conference.

We invite for the following types of submissions (see below for details):

- Paper (oral/poster/demo)
- Performance
- Performance Plus Paper
- Exhibition works
- Installation
- Workshop

Important dates
- SID exhibition proposals: 5 November 2010 (22:00 CET)
- Paper/performance/installation/workshop submission: 31 January 2011 (22:00 CET)
- Review notification: 18 March 2011
- Final paper deadline: 26 April 2011

For any further information/questions/comments/ suggestions, please contact the organizing committee.

Topics
- Novel controllers and interfaces for musical expression
- Novel musical instruments
- Augmented/hyper instruments
- Novel controllers for collaborative performance
- Interfaces for dance and physical expression
- Interactive game music
- Robotic music
- Interactive sound and multimedia installations
- Interactive sonification
- Sensor and actuator technologies
- Haptic and force feedback devices
- Interface protocols and data formats
- Motion, gesture and music
- Perceptual and cognitive issues
- Interactivity design and software tools
- Sonic interaction design
- NIME intersecting with game design
- Musical mapping strategies
- Performance analysis
- Performance rendering and generative algorithms
- Machine learning in performance systems
- Experiences with novel interfaces in live performance and composition
- Surveys of past work and stimulating ideas for future research
- Historical studies in twentieth-century instrument design
- Experiences with novel interfaces in education and entertainment
- Reports on student projects in the framework of NIME related courses
- Artistic, cultural, and social impact of NIME technology
- Biological and bio-inspired systems
- Mobile music technologies
- Musical human-computer interaction
- Multimodal expressive interfaces
- Practice-based research approaches/ methodologies/criticism

Call for papers
We welcome submissions of original research on all above mentioned (and other) topics related to development and artistic use of new interfaces for musical expression. There are three different paper submission categories:

- Full paper (up to 6 pages in proceedings, longer oral presentation, optional demo)

- Short paper/poster (up to 4 pages in proceedings, shorter oral presentation or poster, optional demo)
- Demonstration (up to 2 pages in proceedings)

Please use this template when preparing your manuscript. Submitted papers will be subject to a single-blind peer review process by an international expert committee. All accepted papers will be published in the conference proceedings, under an ISSN/ISBN reference, and will be available online after the conference.

Call for performances

We welcome submission of pieces for three different types of performance venues:

- Concert hall performance
- Club performance
- Foyer "stunt" performance

Any type of NIME performance pieces are welcome, but we would particularly like to encourage the use of motion capture techniques in performance. For this we can make available several different types of motion capture systems (Qualisys, XSens, Optitrack, Mega). Network pieces and mobile music pieces are also encouraged. Within reasonable limits, we may be able to provide musicians to perform pieces. Typical NIME performance pieces last for 5-15 minutes, but shorter and longer performance proposals may also be taken into consideration.

Submitted proposals will be subject to a peer review process by an international expert committee. Documentation of the performances will be available online after the conference.

Call for performance plus paper

To support more cross-disciplinary work between scientific and artistic research, we highly encourage submission of performance pieces related to papers. Here the scientific presentation may be the basis for the artistic presentation, or vice versa.

Submissions within this category will have to be done for both the piece and the paper, with a clear note that paper and piece belongs together. Evaluation will be done on the combined quality of both submissions.

Call for exhibition works

In connection with NIME 2011 an exhibition on sonic interaction design will be curated in collaboration with the EU COST IC0601 Action on Sonic Interaction Design (SID). For the exhibition we are looking for works using sonic interaction within arts, music and design as well as examples of sonification for research and artistic purposes. The exhibition will take place at the Norwegian Museum of Science, Technology and Medicine and run for three months over the summer 2011. We also aim to include works in public spaces to be presented at various locations in Oslo (possibly) for a shorter duration in parallel with NIME.

This is a curated exhibition, and there is a possibility for funding and assistance to be provided for selected artists. Note that there is an early deadline for submissions within this category (5 November). More information about the exhibition, including pictures of the venue, can be found at the SID web site. Any further enquiries concerning the exhibition should be addressed to the curators: exhibition@nime2011.org.

Call for installations

In addition to the SID exhibition, we also call for installations to be presented during the NIME conference only. These may be foyer location installations or room-based installations in connection to the conference venues.

Submitted proposals will be subject to a peer review process by an international expert committee. Documentation of the installations will be available online after the conference.

Call for tutorials and workshops

We call for short (3 hours) or long (6 hours) workshops and tutorials. These can be targeted towards specialist techniques, platforms, hardware, software or pedagogical topics for the advancement of fellow NIME-ers and people with experience related to the topic. They can also be targeted towards visitors to the NIME community, novices/newbies, interested student participants, people from other fields, and members of the public getting to know the potential of NIME.

Tutorial proposers should clearly indicate the audience and assumed knowledge of their intended participants to help us market to the appropriate audience. Workshops and tutorials can relate to, but are not limited to, the topics of the conference. This is a good opportunity to explore a specialised interest or interdisciplinary topic in depth with greater time for discourse, debate, collaboration.

Admission to workshops and tutorials will be charged separately from the main conference. Proposer(s) are responsible for publishing any workshop proceedings (if desired) and should engage in the promotion of their event amongst own networks. Workshops may be cancelled or combined if there is inadequate participation.

Past NIMEs

NIME 2010: University of Technology Sydney, Sydney, Australia
NIME 2009: Carnegie Mellon University, Pittsburgh, USA
NIME 2008: Casa Paganini, Genoa, Italy
NIME 2007: New York University, New York, USA
NIME 2006: IRCAM Centre Pompidou, Paris, France
NIME 2005: University of British Columbia, Vancouver, Canada
NIME 2004: Shizuoka University of Art and Culture, Hamamatsu, Japan
NIME 2003: McGill University, Montreal, Canada
NIME 2002: Media Lab Europe, Dublin, Ireland
NIME 2001: CHI 2001, Seattle, USA

Table of Contents

All lectures and oral presentations are in Auditorium 1 at the University Library (Georg Sverdrups hus) at the University of Oslo. The poster and demo sessions are in the seminar rooms on the 3rd floor in the same building. Please consult the program book for additional information.

Keynote lectures

Paper session A — Monday 30 May 11:00–12:30

Poster session B — Monday 30 May 13:30–14:30

Demo session C — Monday 30 May 13:30–14:30

Paper session D — Monday 30 May 14:30–15:30

Paper session E — Monday 30 May 16:00–17:00

Paper session F — Tuesday 31 May 09:00–10:50

Poster session G — Tuesday 31 May 13:30–14:30

Demo session H — Tuesday 31 May 13:30–14:30

Paper session I — Tuesday 31 May 14:30–15:30

Paper session J — Tuesday 31 May 16:00–17:00

Paper session K — Wednesday 1 June 09:00–10:30

Paper session L — Wednesday 1 June 11:00–12:30

Poster session M — Wednesday 1 June 13:30–14:30

Demo session N — Wednesday 1 June 13:30–14:30

Keynote Lecture 1: Musical Instrument User Interfaces: the Digital Background of the Analog Revolution

Tellef Kvifte
University of Oslo
Department of Musicology
tellef.kvifte@imv.uio.no

ABSTRACT

In this keynote lecture, examples from the development of new user interfaces on free reed instruments and woodwinds in the 19th century are used as a starting point for discussing user interfaces as part of a wider technological, aesthetic and cultural context. The concepts of analog/digital are used to characterize not only the underlying technology, but also aspects of musical parameters and user interfaces, like "discrete" (scale steps; keys) and "continuously variable" (glissandi/vibrato; slides/sliders).

The free reed instruments – the most common of these being the harmonica, accordion and harmonium – were developed with a large number of different user interfaces. Many of these are now obsolete, but many are still surviving. In my lecture I will argue that the control of digital pitch-classes (scale steps) is a central focus in many of these instruments, and also in other instruments developed in this time period and onwards.

It is further argued that there has been a development from a digital pitch-class-oriented culture to a preoccupation with control of analog musical qualities – especially timbre – in the last part of the 20th century. This has been in parallell to changes in the dominating media and technologies for production and distribution of music, and, obviously, in instrument and user interface design.

Thus, musical instrument interfaces, aesthetic preferences, and dominating production technologies can be seen as a system of mutually dependent elements in this development from digital to analog.

Biography

Tellef Kvifte is full professor at Department of Musicology at the University of Oslo. His research interest spans from Norwegian hardanger fiddle music through theory of rhythm to theoretical organology and music technology, and has published internationally in all these areas. His most recent research concerns perspectives on the co-development of music, music tehcnology, notation and concepts of sounds. Kvifte occasionally also appears on the professional World music scene as a musician, and performs tin whistles, hardanger fiddles, saxophones, laptops and a variety of other instruments with confidence. He worked professionally as a television producer before taking up his academic career, and is still a noted record producer.

Figure 1: Tellef Kvifte (Photo: Tom Hatlestad)

NIME'11, 30 May–1 June 2011, Oslo, Norway.

Keynote Lecture 2: Adventures in Phy-gital Space

David Rokeby
http://www.davidrokeby.com
drokeby@sympatico.ca

ABSTRACT

David Rokeby spent the 10 years from 1981 to 1991 gesturing in mid-air and throwing his body against the virtual while creating *Very Nervous System*, an interactive installation which tracks body movement with video cameras and turns the movement into music and/or sound. Developing this work and exhibiting it around the world gave him a wealth of opportunities to experience and observe what happens when we place our bodies at the conjunction of physical and digital spaces. Since the early nineties, he has often returned to this exploration of "phy-gital" experience in a range of video and sound installations, considering this hybrid space as one of the fundamental features of life in a digital culture.

In his presentation, Rokeby will explore characteristics of the experience of phy-gital space, reflecting in particular on how these features affect interactive performance and interactive performers. Then he will present a variety of projects which expand the notion of interactive performance into publicly accessible interactive installations.

Main thrusts of this examination will include the effect of phy-gital space on the interactor's mind and body, virtuosity in the context of interactive interfaces, and the interface as audience.

Biography

David Rokeby's early work *Very Nervous System* (1982-1991) was a pioneering work of interactive art, translating physical gestures into real-time interactive sound environments. It was presented at the Venice Biennale in 1986, and was awarded a Prix Ars Electronica Award of Distinction in 1991. Several of his works have addressed issues of digital surveillance, including *Taken* (2002), and *Sorting Daemon* (2003). Other works engage in a critical examination of the differences between human and artificial intelligence. The *Giver of Names* (1991-) and *n-cha(n)t* (2001) are artificial subjective entities, provoked by objects or spoken words in their immediate environment to formulate sentences and speak them aloud. David Rokeby has exhibited and lectured extensively in the Americas, Europe and Asia. His awards include a Governor General's Award in Visual and Media Arts (2002), a Prix Ars Electronica Golden Nica for Interactive Art (2002), and a British Academy of Film and Television Arts "BAFTA" award in Interactive art (2000).

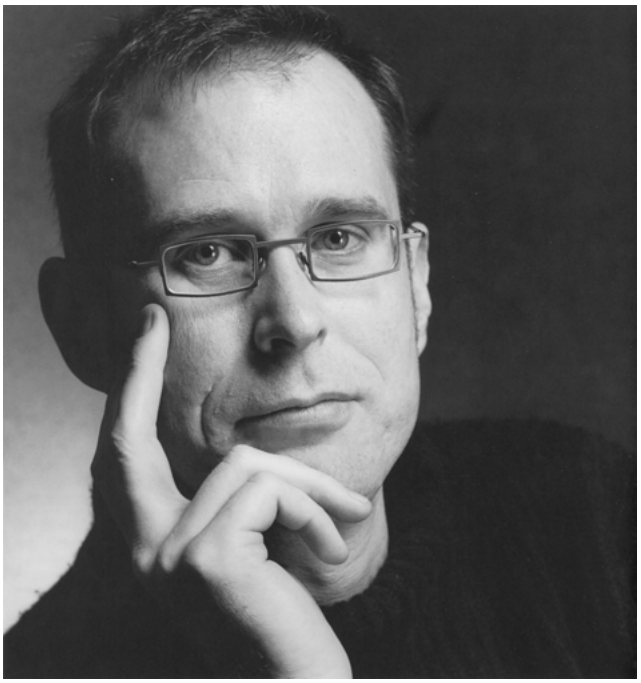


Figure 1: David Rokeby

Keynote Lecture 3: Digital Lutherie and Multithreaded Musical Performance: Artistic, Scientific and Commercial Perspectives

Sergi Jordà
Universitat Pompeu Fabra
Music Technology Group
sergi.jorda@upf.edu

ABSTRACT

In 1982, I was studying for a B.Sc. in fundamental physics, when I first ever saw a computer and soon discovered the magic of computer programming. It was such a revelation that some weeks later I had decided to give up saxophone practice and free jazz in order to become a computer music improviser! Since then, I have pursued the complexity, delicacy and futility of real-time, multidimensional performer–instrument–interaction from different and complementary perspectives.

Initially I did this from a freer and purely aesthetically driven artistic/performer and freelance perspective, then trying to systematize and expand this empirical knowledge from a more scientific/academic point of view as a researcher at the Music Technology Group in Barcelona (1999–present), and more recently, also from an industrial/commercial perspective, manufacturing and selling new electronic musical instruments at Reactable Systems[1] (2009–present).

Art, research and business seem three quite distinct activities, and yet I do not really experience it that way, perhaps because as I understand it, *digital lutherie* cannot work properly without any of these three legs. It has to inevitably start from music, from musical needs and realities, without walking blind or reinventing the wheel at every new step, and at last, without forgetting the potential user.

In this keynote lecture, I will give an overview of my journey from these three angles, with a special focus on what I call "multithreaded musical instruments," the term that could define my main activities and research area for the last 15 years.

Biography

Sergi Jordà holds a B.S. in Fundamental Physics and a Ph.D. in Computer Science and Digital Communication. He is a researcher in the Music Technology Group of Universitat Pompeu Fabra in Barcelona, and a lecturer in the same university, where he teaches computer music, HCI, and interactive media arts. He has written many articles, books, given workshops and lectured though Europe, Asia and America, always trying to bridge HCI, music performance and interactive media arts. He has received several international awards, including the prestigious Ars Electronica's Golden Nica in 2008. He is currently best known as one of the inventors of the Reactable, a tabletop musical instrument that in 2007 accomplished mass popularity after being integrated in Icelandic artist Björk's Volta world Tour.

Figure 1: Sergi Jordà

[1] http://www.reactable.com

The Overtone Fiddle: an Actuated Acoustic Instrument

Dan Overholt
Department of Architecture, Design
and Media Technology
Aalborg University, Denmark
Niels Jernes Vej 14, 3-107
dano@create.aau.dk

ABSTRACT

The Overtone Fiddle is a new violin-family instrument that incorporates electronic sensors, integrated DSP, and physical actuation of the acoustic body. An embedded tactile sound transducer creates extra vibrations in the body of the Overtone Fiddle, allowing performer control and sensation via both traditional violin techniques, as well as extended playing techniques that incorporate shared man/machine control of the resulting sound. A magnetic pickup system is mounted to the end of the fiddle's fingerboard in order to detect the signals from the vibrating strings, deliberately not capturing vibrations from the full body of the instrument. This focused sensing approach allows less restrained use of DSP-generated feedback signals, as there is very little direct leakage from the actuator embedded in the body of the instrument back to the pickup.

Keywords

Actuated Musical Instruments, Hybrid Instruments, Active Acoustics, Electronic Violin

1. INTRODUCTION

The Overtone Fiddle follows upon the development of the author's prior Overtone Violin [6], with a change of focus towards another area of investigation. Whereas the Overtone Violin is entirely electronic (there is no use of a resonating acoustic body), the Overtone Fiddle described here integrates a full acoustic chamber. It receives resonant stimulation directly from both the strings on the instrument, and from an internally mounted tactile sound transducer, which is controlled via DSP running on an attached iPod Touch®. The physical design of the instrument accommodates this by incorporating space to mount the iPod locally, as can be seen in Figure 1.

The instrument is essentially a 'pochette' [13] type of violin design, with a standard violin length and a 2" external width. Luthier Don Rickert, of Adventurous Muse [9] made plans and constructed the instrument requested for this project. The internal cavity is 1.75" wide in order to accommodate a specific tactile sound transducer. The overhanging top and back, in addition to adding to the vibrating surface, and thus, sonority of the instrument, also provide mounting surfaces for external components such as batteries, pickup preamps, and so forth. The entire back of the instrument is made of maple, and screwed onto the instrument sound chamber (main body) in order to allow easier access to internal elements and wiring.

2. SENSOR & ACTUATOR DESIGN

The Overtone Fiddle uses a tightly focused sensing approach to capture string vibrations – several designs were attempted, only the chosen method is described herein.

Figure 1. The Overtone Fiddle – first prototype.

2.1 Magnetic Pickup

While an optical pickup system similar to that used with the Overtone Violin could have been designed for the Overtone Fiddle as well, it was deemed unnecessary, as a commercially available magnetic pickup system was found that captures the movements of the strings themselves directly. This system is called REBO [8], and it functions in the same manner as an electromagnetic guitar pickup. It is mounted to the end of the instrument's fingerboard (see Figure 2, left).

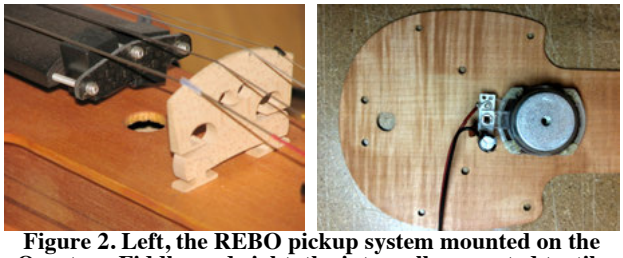


Figure 2. Left, the REBO pickup system mounted on the Overtone Fiddle, and right, the internally mounted tactile sound transducer (not shown here, a similar transducer is located in the instrument's second acoustic body).

2.2 Tactile Sound Transducers

Signals can be injected directly into the main acoustic body of the instrument via a voice-coil type of tactile sound transducer (see Figure 2, right), as well as into an optional second acoustic body that hangs below the instrument (see Figure 3). This lower resonating body is made of very thin carbon fiber and balsa wood – materials that would not be strong enough to support the full string tension of strings on the main body – thus allowing extremely efficient transfer of acoustic energy from another embedded tactile transducer, to the structural elements of the box. The box itself is quite a simple design for this prototype. It was designed to dimensions allowing it to function as a proper Helmholtz-resonator (internal shape and porting), in order to maximize the volume of the resulting sound. As such, it is actually capable of producing louder tones than the main body of the instrument.

Figure 3. The Overtone Fiddle with carbon fiber / balsa wood second acoustic body mounted underneath.

Designed as a 5.6" x 5.6" x 1.2" box, this second body has a total internal volume of roughly 37.6 cubic inches. To relate this to the spherical shape of a traditional Helmholtz resonator, solving equation 1 below provides the radius of a sphere with an equivalent internal volume. Then, to arrive at the proper size of the soundhole, this radius is divided by four, resulting in a prototype design with a 0.519" radius circular soundhole.

$$R = \sqrt[3]{\frac{V}{4/3\ PI}}$$

Equation 1. Solving for the radius of an equivalent spherical Helmholtz resonator, given the internal volume of the prototype second body.

The second body of the Overtone Fiddle is also driven with DSP-generated feedback signals, usually based on the sound from the strings (indeed, any audio signal the performer desires is possible, if suitably programmed). Many types of responsive software can be programmed to run on the iPod touch mounted on the fiddle. Sounds and effects can be responsive to any motions sensed by the accelerometers and gyroscopes in the iPod Touch, with parameters controlled by both real-time analysis of incoming sound from the strings, and gestural movements of the performer.

The tactile transducers inside both the top and bottom resonating bodies are driven independently by a 11.1volt Li-Ion battery-powered Class-T stereo audio amplifier. As mentioned, the main body of the instrument is designed to accommodate this, by providing space for mounting the battery, amplifier, and associated circuitry. This makes the entire instrument self-contained (not including the bow and its corresponding electronic circuit).

2.3 Bow Design

The bow used with the Overtone Fiddle is custom made by the E.W. Incredibow company, from a simple carbon fiber rod that is lighter (almost half the weight of a wood bow) and longer than a normal violin bow. This is helpful in order to accommodate the added mass of a small battery-powered sensor circuit based on the CUI32 [5], along with a wireless 802.11g radio module [11], and an absolute orientation sensor [2]. A simple BASIC-language program was written on the CUI32 using StickOS [12], which is the default operating system for the CUI32. It receives the stream of orientation vectors from the sensor, and translates them into Open Sound Control (OSC) protocol, in order to send them to the iPod Touch. The orientation sensor reports the direction in which the bow is pointing using Euler angles or quaternions, sending updates at 300Hz. This is accomplished by its sensor fusion algorithm, which combines data from internal accelerometers, gyroscopes, and magnetometers.

The WiFly module is configured to broadcast its own 'AdHoc' 802.11 base station, which is then chosen as the network to join in the 'WiFi settings' of the iPod Touch. This allows the CUI32 to send UDP and/or OSC and communicate easily with the iPod Touch. One of the strengths of this approach is that the orientation of the bow can be compared to the violin/iPod's orientation (as determined using the iPod's internal sensors) and differences in these measurements can be used to control various parameters of real-time effects processes. The mapping of such controls to real-time parameter updates is of course a major task given to the composer / performer / programmer of the system.

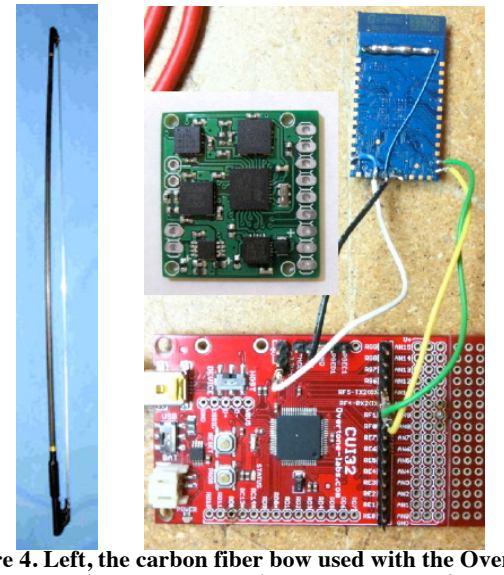


Figure 4. Left, the carbon fiber bow used with the Overtone Fiddle, and right, the electronics components before being attached to the bow (red CUI32 bottom, blue 802.11g radio module top, and green absolute orientation sensor middle).

3. MUSICAL PROGRAMMABILITY

Software used with the Overtone Fiddle can be written in a variety of applications, such as SuperCollider [4], PureData (using libpd [3] or RjDj [10]), or the MoMu the Mobile Music Toolkit [1], all of which can run in real-time on the iPod Touch. In general, the author tends to use SuperCollider, but has experimented extensively with others as well. The instrument can thereby incorporate all of the flexibility of modern digital signal processing techniques, for example, altering the timbral structure of the sound being produced in response to player input.

In order to use the sensor data from the bow with any of these iPod Touch applications, the data must be formatted into either OSC as mentioned earlier, or another network format that the iPod application can receive via the WiFly module. For an example here, Figure 5 shows BASIC code that captures sensor values on the 16 analog input pins on the CUI32, and sends them as UDP to a custom "scene" in RjDj – really just a PureData patch, which is shown in Figure 6. In this case, the absolute orientation sensor was not used, as this was a simple test to verify connectivity. Nonetheless, there are many different types of analog sensors that can be used with the 16 analog input pins on the CUI32. Therefore, such a setup will almost surely be useful in the future.

To explain the BASIC code shown in Figure 5, it can be seen that on line 10 the 2nd UART (serial port of the CUI32) is initialized. This UART is connected to the WiFly module via two wires: one for transmitting, and one for receiving. Lines 20 through 160 are declaring variables "a … p" that correspond to the 16 analog input pins on the CUI32. These are configured as 'debounced', which causes a simple 3-point running average filter to be executed on the incoming sensor values, in order to smooth out any glitches. Lines 170-210 create a connection

between the iPod Touch and the CUI32 (the iPod must already have joined the WiFly's AdHoc network).

```
10  configure uart 2 for 115200 baud 8 data no parity
20  dim a as pin an0 for analog input debounced
30  dim b as pin an1 for analog input debounced
40  dim c as pin an2 for analog input debounced
50  dim d as pin an3 for analog input debounced
60  dim e as pin an4 for analog input debounced
70  dim g as pin an6 for analog input debounced
80  dim h as pin an7 for analog input debounced
90  dim i as pin an8 for analog input debounced
100 dim j as pin an9 for analog input debounced
110 dim k as pin an10 for analog input debounced
120 dim l as pin an11 for analog input debounced
130 dim m as pin an12 for analog input debounced
140 dim n as pin an13 for analog input debounced
150 dim o as pin an14 for analog input debounced
160 dim p as pin an15 for analog input debounced

170 sleep 100 ms  rem -- wait for WiFly to boot
180 print "$$$"; rem -- tell WiFly to enter config mode
190 sleep 300 ms  rem -- wait for it to enter config mode
200 print "open 169.254.1.3 2000"  rem -- connect to iPod Touch
210 sleep 100 ms  rem -- wait for connection to establish

220 configure timer 0 for 10 ms  rem -- interrupt update rate 100Hz
230 on timer 0 do print "A",a,b,c,d,e,"0",g,h,i,j,k,l,m,n,o,p,";"
240 while 1 do  rem -- no need to do anything in the main loop because
250 endwhile  rem -- everything is handled by the interrupt routine
```

Figure 5. StickOS BASIC program that sends the CUI32's analog sensor inputs to the [netreceive] object in the RjDj app on the iPod (running a corresponding RjDj "scene", which is actually the PureData patch shown in Figure 6).

Finally, line 220 enables an internal timer in the CUI32 (functionally similar to the [metro] object in PureData), and configures it to cause events every 10 milliseconds. Every time one of these events happens, line 230 sends the actual sensor values to RjDj, which is running the PureData patch seen in Figure 6. The list of sensor values is always preceeded with a capital "A", and appended with a semicolon. In PureData, the semicolon is needed by the [netreceive] object to signify the end of a packet, and the capital "A" is used as an identifier to signify the beginning of the packet. The top [match] object in Figure 6 always checks for the capital "A" (number 65 in ASCII-code) so that synchronization is maintained.

The same functionality can be achieved with other iPod applications with a few modifications to the code. For example, SuperCollider requires the use of OSC-format strings in order to receive network data, so the addition of proper OSC syntax (string identifiers and 4-byte boundaries) is added to the BASIC code in order to use it with SuperCollider running on the iPod Touch. For the sake of brevity, an example of such is not included here.

The CUI32 circuit board was designed by the author as an improved version of the CREATE USB Interface (CUI) [7], and is sold by SparkFun electronics and other online retailers.

4. NEW PERFORMANCE TECHNIQUES

Since any DSP algorithms in use can be controlled through gestural interaction using both the motion sensors in the iPod (accelerometers, gyroscopes, etc.) as well as the electronics on the bow, the system promotes new performance techniques for interactions above and beyond those supported by traditional acoustic instruments. For instance, in initial improvisational performances by the author, the timbre of the Overtone Fiddle changes is made to change dramatically when rapid movements are performed.

The multitouch screen surface on the iPod is also useful in certain musical contexts. While it clearly cannot be used while simultaneously bowing anything other than open strings, there can nonetheless be sections of a performance allowing the performer enough time to access the screen. For example, a DSP algorithm can sustain a note beyond the time that the performer actually bowed it, thus allowing modifications to the timbre thereafter through interaction with the multitouch screen. Simple parameter changes can of course be executed in between notes as well, etc.

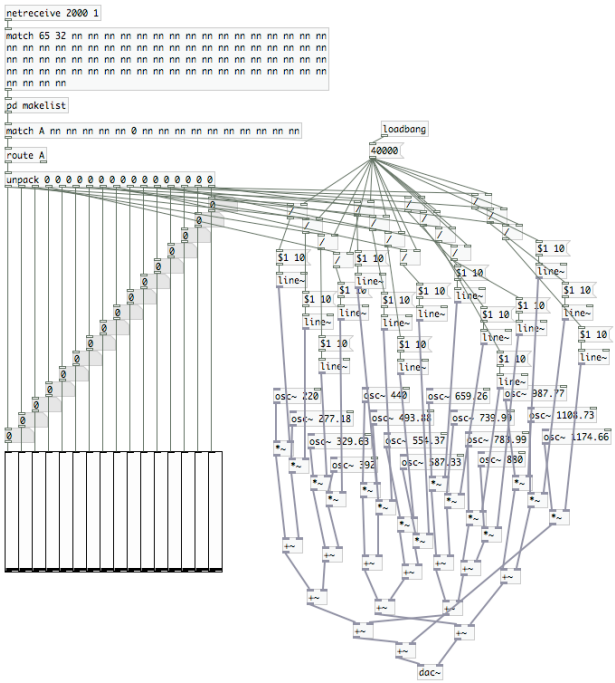


Figure 6. PureData patch used in RjDj to receive real-time analog sensor data from the CUI32. On the left side, sliders visually represent incoming sensor values, and on the right side, a simple additive synthesizer used for testing that generates sounds in response to the sensor data.

4.1 New Sonic Possibilities

DSP algorithms are used to adjust the body vibrations of the acoustic part of the instrument, actively changing the harmonics heard in the musical tone quality. Consequently, the acoustic properties of the Overtone Fiddle are adjustable, rather than being permanently defined by the woodworking skills of a luthier. In other words, the internal actuator can cause the wooden body to exhibit new dynamic behaviors, and it is this methodology that distinguishes the Overtone Fiddle from prior instruments such as the Chameleon Guitar [14].

While the sound quality of a traditional acoustic instrument is fixed by its physical design, actuated musical instruments can malleably simulate even physically impossible acoustic properties, such as a violin with hundreds or thousands of sympathetic strings; this would be akin to extreme versions of instruments like the Norwegian Hardanger Fiddle, or the Viola d'Amore from the baroque period. For the performer and the audience, however, the perception is that the sound being produced by an actuated acoustic musical instrument such as the Overtone Fiddle is somehow physical – the resulting music is created through gestures on the instrument held close to the performer, who can use the technology to produce interesting timbres that might never have been heard before. It is an important consideration for the performer, that the Overtone Fiddle is not connected by cables to a computer, nor to any remote loudspeakers.

5. MUSICAL OBJECTIVES

The main objective of the development of the Overtone Fiddle is to pursue a long-term research project that is focused on the

development of new acoustic musical instruments – at first, in the bowed string family and then expanding to others. The addition of technological components to acoustic instruments is used in order to extend these existing instrument types with new expressive and performative possibilities. The project also aims to explore the potentials that these instruments have in both new compositions and new methods of performance.

It is the author's hope that the development of such new instruments will help revive the evolution of more traditional musical instruments today, through a combination of some of today's most advanced technologies with traditional instrument making and musical skill and practice. In this sense, traditional acoustic instruments are already seen as advanced technologies in their own right, because they have been refined over many years of development. The goal is to add new dimensions and expressive possibilities to the capabilities of traditional acoustic instruments, and to explore these in contemporary music and performance. The research project should not be construed as an attempt to improve the basic acoustic instrument designs themselves, but seen as an extension of said instruments' expressive and performative range.

Because the acoustic properties of the Overtone Fiddle can be changed through mathematical processes in real time, it allows artists to make radical changes to their sound. This gives an opportunity to create music that explores new sound worlds, yet sill follows in the traditional musical training to a certain degree. Composers and performers can make use of the instrument's programmability by means of sound synthesis, sound effects and generative algorithms, all of which can be configured to respond to input from the musician and allowing an almost infinite number of different instrument interaction methods.

6. CONCLUSION AND FUTURE WORK

The development of the Overtone Fiddle offers both technical and artistic challenges that the author enjoys embracing. It has been shown that the first prototype of the instrument is capable of many new musical interactions – future versions of the fiddle, as well as other bowed string instruments are currently in the works. While cables or even wireless audio transceivers could be used to enable the Overtone Fiddle to connect to a laptop for more powerful signal processing than is possible with an iPod Touch, keeping the instrument compact and self-contained is a high priority for this project. Nonetheless, initial experiments were done using a laptop, and OSC-sending remote-control apps such as TouchOSC or Fantastick.

Future prototypes of actuated bowed string instruments may incorporate more traditional violin bodies, instead of the 'pochette' type of design. They may also involve a new concept developed in this research project, that of a "bridgeless" violin that was tested in the process of building this first prototype. In this case, an extended fingerboard is used, with the wide end culminating in a raised nut where the bridge would normally be. The strings are then entirely supported by the fingerboard (never touching the body of the instrument). The instrument body is hung from a bracing system running underneath (that also supports the iPod and electronics), in order to separate the fingerboard vibrations from the actuated body. With this setup, a secondary instrument body is not used.

The author can be seen playing the prototype Overtone Fiddle in a video titled "An Evening of Actuated Instruments" together with Edgar Berdahl on the Haptic Drum and Robert Hamilton on the Feedback Resonance Guitar in an improvisational setting. This video is located online at the Actuated Musical Instruments Guild website: http://actuatedinstruments.com/.

7. ACKNOWLEDGEMENTS

I would very much like to thank Chris Chafe, Max Mathews, Edgar Berdahl, Rich Testardi, Don Rickert, and my wife Anne-Marie Hansen for all of their help with this project. The support they have given me has been a great help in many ways.

I also wish to acknowledge Lars Beer Nielsen at the Innovation Center Denmark, and Keith Devlin at Stanford University's Human Sciences & Technologies Advanced Research Institute (H-STAR) for their financial support, providing the opportunity to spend 2 months at Stanford's Center for Computer Research in Music and Acoustics (CCRMA) working on this research.

8. REFERENCES

[1] Bryan, N. J., Herrera, J., Oh, J., Wang, G. Momu: A mobile music toolkit. *In Proceedings of the. International Conference on New Interfaces for. Musical Expression (NIME)*, Sydney, Australia, 2010.

[2] CH Robotics, http://www.chrobotics.com/ accessed January 29, 2011.

[3] LibPd, http://gitorious.org/pdlib/ accessed January 29, 2011.

[4] McCartney, J. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal, 26(4), 61-68.* 2002

[5] Overholt, D. CUI32 microcontroller board, http://code.google.com/p/cui32/ accessed January 29, 2011.

[6] Overholt, D. The Overtone Violin: a New Computer Music Instrument. *Proceedings of the International Computer Music Conference*, *ICMC 2005,* (Barcelona, Spain, 5-9 September 2005).

[7] Overholt, D. Musical interaction design with the CREATE USB Interface: Teaching HCI with CUIs instead of GUIs. *Proc. of the 2006 International Computer Music Conference*, New Orleans, 2006.

[8] REBO, http://www.uli-boesking.de/rebo/ accessed January 29, 2011.

[9] Rickert, D. http://www.adventurousmuse.com/ accessed January 29, 2011.

[10] RjDj, http://www.rjdj.me/ accessed January 29, 2011.

[11] Roving Networks (WiFly GSX module), http://rovingnetworks.com/ accessed January 29, 2011.

[12] Testardi, R. (StickOS), http://cpustick.com/stickos.htm accessed January 29, 2011.

[13] Pochette (kit violin), http://en.wikipedia.org/wiki/Kit_violin accessed January 29, 2011

[14] Zoran A. and P. Maes, Considering Virtual and Physical Aspects in Acoustic Guitar Design, *Proceedings of New Instruments for Musical Expression (NIME) Conference*, Genova, Italy, June 5-7, 2008.

A Low-Cost, Low-Latency Multi-Touch Table with Haptic Feedback for Musical Applications

Matthew Montag, Stefan Sullivan, Scott Dickey, and Colby Leider
Music Engineering Technology Group
University of Miami
Frost School of Music
{matt.montag, stefan.sullivan}@gmail.com, d.dickey@umiami.edu, cleider@miami.edu

ABSTRACT

During the past decade, multi-touch surfaces have emerged as valuable tools for collaboration, display, interaction, and musical expression. Unfortunately, they tend to be costly and often suffer from two drawbacks for music performance: (1) relatively high latency owing to their sensing mechanism, and (2) lack of haptic feedback. We analyze the latency present in several current multi-touch platforms, and we describe a new custom system that reduces latency to an average of 30 ms while providing programmable haptic feedback to the user. The paper concludes with a description of ongoing and future work.

Keywords

multi-touch, haptics, frustrated total internal reflection, music performance, music composition, latency, DIY

1. INTRODUCTION

1.1 Motivation

Multi-touch input devices have the potential to serve as highly expressive musical instruments. The plurality of potential control inputs (e.g., position, relative distance, relative rotation, etc.) and the simultaneity of these inputs provides considerable control of parameters compared with many current electronic or acoustic instruments. As a performance device, this provides great potential for interesting and dynamic audio/musical control. Additionally, the large size of many multi-touch tables accommodates larger gestural movements, helping the performer to physically interact with the music, as well as allowing for collaboration and interaction among multiple users simultaneously.

A multi-touch table surface can also be used as a video projection surface, providing visual feedback to the performer without blocking line-of-sight with the audience. In addition to enhancing the performer's experience, multi-touch tables allow audiences to witness performance gestures, providing increased emotional connection between performer and audience [20]. Recent advances in a number of technologies have also added to the growing do-it-yourself (DIY) multi-touch movement [21], a design philosophy under which the work described here falls.

However, camera-based multi-touch tables suffer from input lag and event quantization imposed by the camera's frame-capture rate. The input lag may vary based on idiosyncrasies of the camera and host computer configuration. For example, if JPEG frame compression is performed on the camera in order to transfer a 640×480 image at 30 frames per second (FPS) over a USB 1.0 connection, the frame must be decompressed by the host computer before it is further processed, adding a significant delay to the input path. This motivates an investigation of camera selection and host configuration in the pursuit of minimum latency. Additionally, the table interface provides minimal haptic feedback—only the sensation of touch that occurs when a finger is physically in contact with the table's surface.

1.2 Background

Optical-based touch surfaces use infrared light and an IR-sensitive camera to detect the disturbance caused by a touch event. Two common methods are laser light plane illumination (LLP) [22] [18] and frustrated total internal reflection (FTIR) [7]. The LLP method uses infrared lasers to create a laser plane a few millimeters above a translucent touch surface. When users touch the surface, their fingers are illuminated by the light plane and become visible to the camera below. Because the laser plane is situated above the touch surface, the user is able to create a touch event without actually touching the table. FTIR touch surfaces operate by shining infrared light into the edge of a sheet of acrylic. The light is internally reflected by the acrylic, and it escapes only when diffused at the contact area between the surface and the user's finger.

The current project was informed by lessons learned from our first attempt at a multi-touch table (Figure 1) [18], which was constructed using the laser light plane method. The FTIR and LLP methods are similar in cost and operation, but because the FTIR system registers a touch exactly when the user touches the surface, this method is well-suited for percussive musical interaction under strict latency constraints.

1.3 Multi-Touch Latency, Audio, and Haptics

Audio latency can prove a significant impediment to musical performance [25] [16] [6], and multi-touch surfaces are particularly prone to relatively large latency times, as discussed above. In the realm of music, many authors agree that latencies of up to 30 ms between gestural input and sonic result are allowable in most real-time performance situations [14] [17].

Other studies discuss the perception of haptic-audio asynchrony (e.g., [1] [24]) and audio-visual-haptic asynchrony (e.g., [8]). One study notes that haptic-audio asynchrony can be detected with latencies of as little as 2 ms [24] with respect to event time-order, while others [15] [1] note just-noticible differences (JNDs) of 18–42 ms to haptic-audio events. These numbers give us a reasonable target. The

audio latency of our first-generation system was over 100 ms, giving us much room for improvement in building a new table.

2. OUR MULTI-TOUCH TABLE

In this project, we attempted to leverage an FTIR-based optical tracking surface with a high-frame-rate camera. The system output that drives the audio display is used to simultaneously drive the haptic display, resulting in no latency between audio and haptic feedback systems.

2.1 System Overview

Incorporating lessons learned from our previous multi-touch table and recent literature, we constructed a second-generation table. The table surface is 90×67 cm and was constructed using off-the-shelf materials for around US$ 300, plus the cost of a short-throw projector that drives the display (Table 1). We edge-light a 3/8" acrylic touch surface with a strip of 850 nm infrared LEDs. The surface is overlaid with a sheet of silicone-treated vellum, which serves as a projection surface and compliant layer. The compliant layer helps touches appear brighter to the camera underneath. Our new table incorporates a PlayStation Eye camera running at 100 FPS. Community Core Vision (ccv.nuigroup.com) is used to process the video input and generate TUIO (tangible user interface object) messages [13]. Max/MSP or Processing receives the TUIO messages, and this software layer performs event logic for turning these messages into audio output. Other software configurations are the subject of ongoing research. The host computer is connected to a multichannel audio interface that drives a 10.2-channel audio display comprised of ten Genelec 8020A active loudspeaker monitors and two Genelec 7050B active subwoofers.

2.2 Adding Haptic Feedback

Haptic feedback can be used to provide an additional dimension of feedback for the performer on a multi-touch table. The inclusion of haptic feedback has been shown to increase performance accuracy significantly [19], and many recent musical instruments engage haptics as a central feature of their interface [4] [10] [5] [3]. Haptic interaction in multi-touch tables has previously been described in the context of physical "pucks" that a user places on the table [23] [12]. More recent haptic multi-touch displays use electrical fields [2], pnematic pressure [9], and magnetic fields [11] to provide tactile feedback. Our implementation is unique in that we couple a single large tactile transducer capable of producing up to 20 foot-pounds (89 N) of force onto the surface of the table itself, which can achieve a variety of tactile effects ranging from subtle to startling.

Table 1: Construction Costs

Component	Cost (US$)
Polished Acrylic	$100
LED Strip (2 m)	$96
Compliant Surface	$20
Wooden Frame	$25
Short-Throw DLP Projector	$400
Playstation 3 Eye Camera	$27
Power Supply	$16
Amplifier	$25
Actuator	$10
Total	$719

We found that the most efficient method of vibrating the multi-touch surface was to couple a tactile transducer (the Aura AST-1B-4) directly to the surface. We had tested haptic feedback with a DC motor driven by a microcontroller that was connected to the host computer via USB, but this configuration exhibited noticeable latency. On the other hand, the tactile transducer which is driven by an audio signal provides near-zero latency and guarantees that haptic events are synchronous with audio output.

The question of when and how to provide haptic feedback to the user is a function of the software application, and may vary from a simple touch response to a method of indicating perceived edges or proximity when the user drags a finger across a virtual surface boundary. For instance, our first approach was to generate a short transient "bump" on the surface every time a touch event occurred.

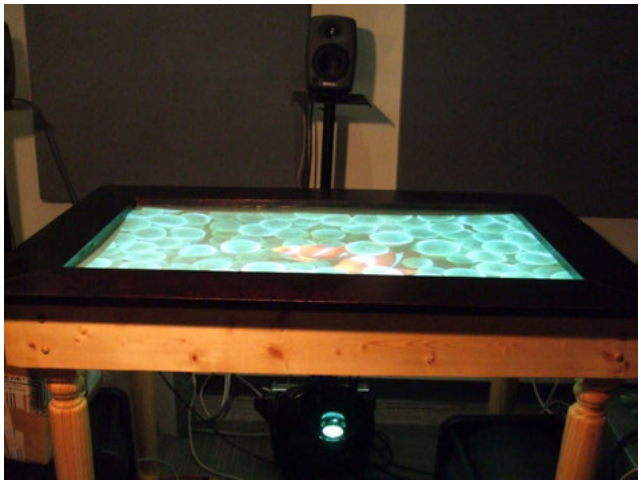


Figure 1: Our first-generation multi-touch table, using the Laser-Light Plane (LLP) method.

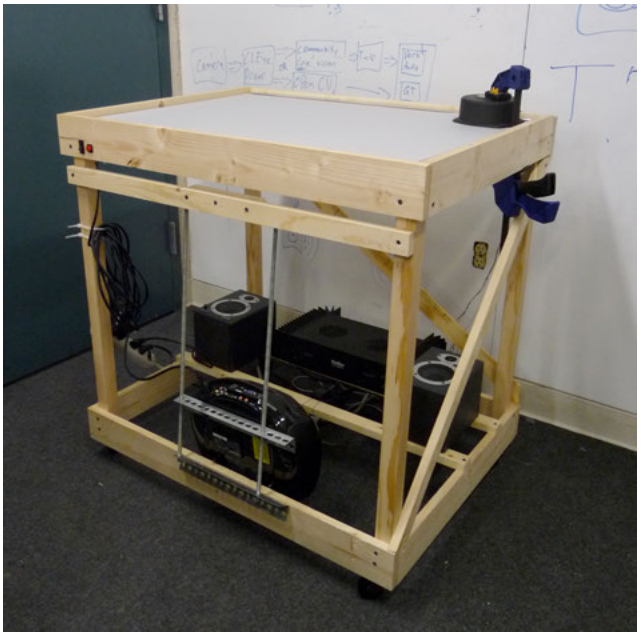


Figure 2: Second-generation multi-touch table, incorporating FTIR sensing, haptic feedback, and a simplified two-channel audio display used during development. Note the transducer coupled directly to the surface of the table at the top right.

In terms of user experience, this provides only marginally more information than the mechanical feedback of touching a motionless table, but informal tests suggest this improves users' perception of multimodal simultaneity. We describe ongoing efforts to incorporate haptic feedback into a musically expressive device later in this paper.

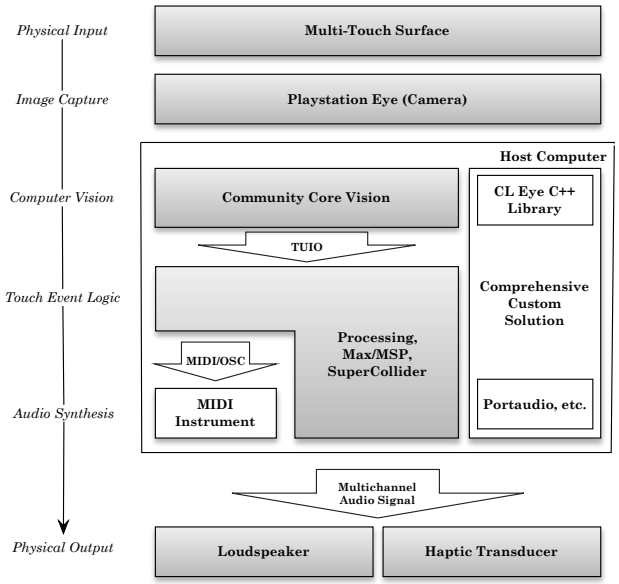


Figure 3: System-level overview of our multi-touch table. Boxes in gray are currently implemented, and boxes in white are potential altertnatives.

3. LATENCY TESTS

To assess the audio and video latency present in this new multi-touch table, a variety of tests were performed on the table and other existing systems for comparison.

3.1 Video Latency Tests

Video tests were performed prior to audio latency tests to determine the delay contributed by different cameras and display devices. The test was carried out in the following manner. The camera and display device under question were connected to the host computer. A frame-counter window and camera monitor window were shown side-by-side on the display device. The camera was aimed at the display so that both the frame counter and the video-overlay window were visible, creating a video loop. It was then possible to photograph the display, recording the counter and a time-delayed version of the same counter in one image, allowing a simple subtraction of these two counters to determine the video latency. We tested the Unibrain Fire-i firewire camera (maximum 30 FPS) and the Sony PlayStation Eye USB 2.0 camera (maximum 100 FPS) with a Toshiba TDP ET-10 DLP projector (60 Hz refresh rate), a Dell E173FP LCD display (60 Hz), and an E-machines CRT monitor (100 Hz). The results were averaged over a minimum of 25 measurements. This testing setup is shown in Figure 4.

3.2 Video Latency Results

The results of the video latency tests showed that the CRT was the fastest display device, and the PlayStation Eye was the fastest capture device. The results can be seen in Figure 5. The Unibrain Fire-i at 30 FPS and PlayStation Eye at 100 FPS resulted in a video loop latency of 70 ms and

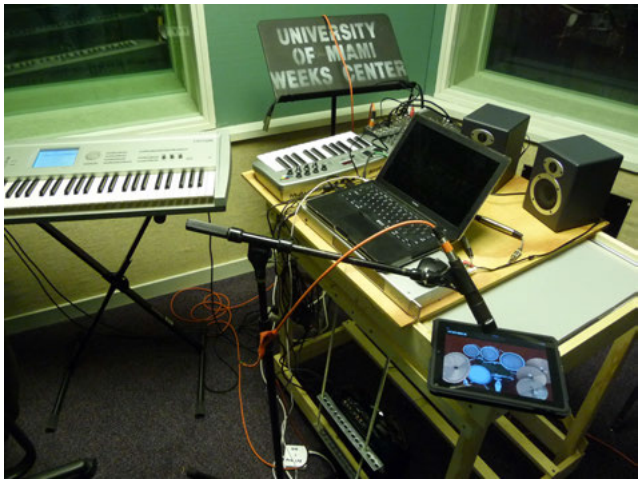


Figure 4: Experimental setup for testing latency times for our second-generation multi-touch table and other devices.

10 ms, respectively. It is clear that the Unibrain Fire-i suffers more input lag than can be explained by the frame rate alone, which accounts for only 23 ms of the difference. Although the display device itself does not impact latency of the audio output, it is important to note the large disparity in latency among the tested displays. The DLP projector exibited a video input lag around 80 ms. This delay is imparted by a particular digital image-processing circuit in this Toshiba projector and is not inherent to all projectors or DLP technology.

3.3 Audio Latency Tests

We measured the delay between input touch event and audio output of our low-latency multi-touch table configuration and benchmarked our system against several other multi-touch and MIDI devices. The following devices were tested: Korg Triton keyboard, USB MIDI keyboard, Apple iPad, Apple iPod Touch, HTC Hero Android-based smartphone, and our new multi-touch table. Multi-touch and MIDI tests were conducted with an Intel Macbook 2.2 MHz Core 2 Duo running Windows XP, connected to an Echo AudioFire 12 firewire audio interface configured with a 256 sample buffer at 44.1 KHz. Apple iPad tests were performed with the apps Drum Kit Pro and I Can Drum. The Apple iPod was tested with Drum Kit Pro; the Android phone was tested with DrumKit.

In all test cases, we placed a microphone near the control surface and recorded a stereo audio file, with the touch/ keypress noise on the left channel and the corresponding audio output on right channel. We then used a sound file editor to measure the time elapsed between the trigger event and the sound output, averaging at least 25 trials. In the case of the keyboards, the impact of the key on the keybed was chosen to mark the trigger event. This process is illustrated in Figure 6.

3.4 Audio Latency Results

The results of the audio latency tests show that the Korg Triton had latency between 0 ms and –1 ms. Note that this is a valid result given the testing method, although it indicates that the electrical contact for the Triton's keys occurs somewhere before the fully depressed position. The Triton has a known latency of less than 2 ms, and our result is included as a reference point for the test method. The Android and Apple devices were shown to have average laten-

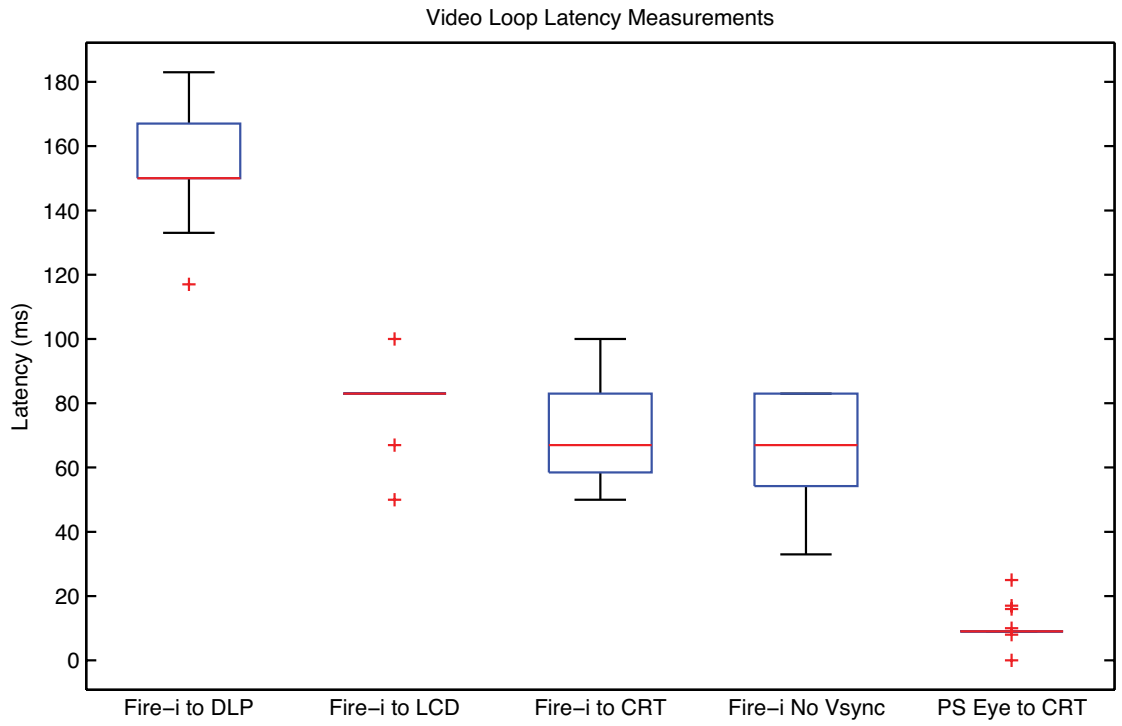


Figure 5: Results of video latency tests.

cies above 50 ms, depending on the application. The multi-touch table with Fire-i camera exhibited a higher latency than the Apple iPad and Apple iPod Touch. The measured latency of the multi-touch table using the PlayStation Eye was much lower and surprisingly comparable to the USB MIDI keyboard. Our average overall latency for the table was 30 ms. These results are summarized in Figure 7.

4. ONGOING AND FUTURE WORK

Our team is currently working on several improvements and related projects. These include custom music applications that take advantage of haptic multi-touch interaction, improvements to the table itself, assessment and improvement of the table's haptic feedback system, and other long-term projects. Each of these is described below.

4.1 Software Applications

Our primary ongoing project is to create software applications that take advantage of the multi-touch table as a musical performance instrument. We emphasize applications that use multi-touch input in such a way that could not be easily duplicated by point-and-click interaction, for example, the simultaneous manipulation of several control points along a virtual vocal tract, or the simultaneous control of several harmonic overtones in an additive-synthesis instrument. We also wish to take advantage of our table's low-latency characteristics and we are creating responsive virtual instruments for live performance.

4.2 Table Improvements

As seen in the video latency results above, there is considerable room for improvement in the high latency of the tested DLP projector. The low-latency audio signal should be accompanied by low-latency visual feedback. We are actively looking for low-cost short throw projectors with exceptional latency characteristics.

We are also currently working by trial and error to improve the performance of the compliant surface used in our FTIR table. Silicone applied to the bottom of the vellum occasionally sticks to the acrylic surface in the area of a touch, preventing the recognition of subsequent touches in that area. This deficiency impacts musical performance, but could be resolved by incorporating the laser light plane method in future experiments.

4.3 Assessment and Improvement of the Haptic Subsystem

The table described here delivers a transient vibration on its surface each time a user touches it, and, as mentioned, this anecdotally improves perception of event simultaneity. A more meaningful implementation might be to provide varying degrees of haptic feedback as a musician drags a finger across the table, to indicate certain particular tasks/events. For instance, the table might vibrate every time a certain sound event begins and/or terminates, or every time a finger moves from one parameter to another, or indicating certain thresholds of parameters/events. We are also investigating haptic feedback to impart additional information about a sound as it is being "scrubbed." For example, search time when dragging across a graphical waveform representation of a long-duration sound file to find a particular section may be minimized by using haptic displays of simulated surface texture and intensity to represent a particular feature vector, e.g., local novelty. Finally, we are also investigating the creation of physical "buttons" on our multi-touch surfaces by using the haptic subsystem to generate Chladni patterns on the surface of the acrylic.

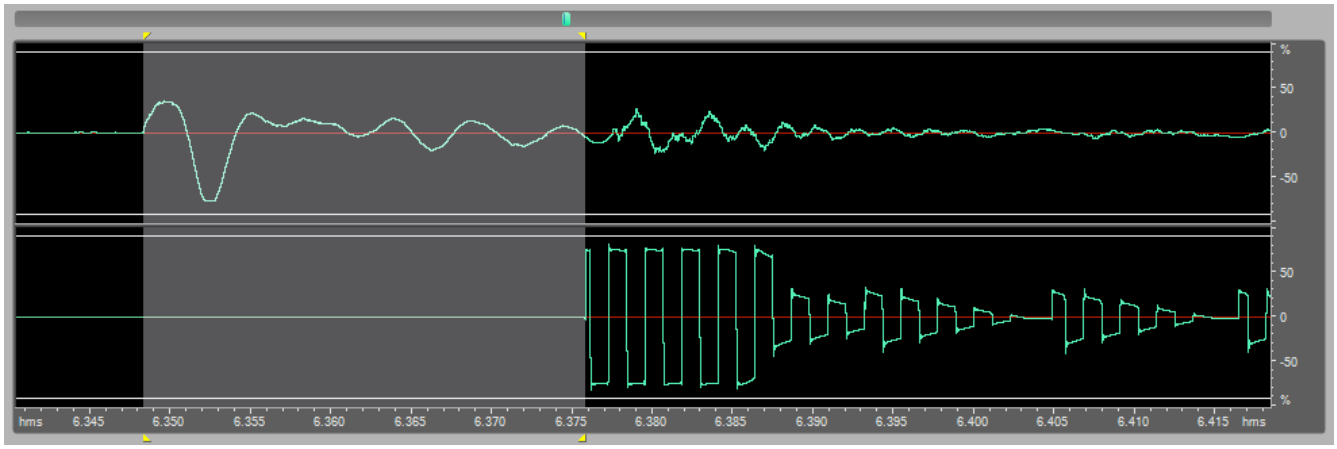


Figure 6: Measuring a 27-ms duration between a finger tap on the table surface, recorded in the left channel, and the synthesized sound output, recorded in the right channel.

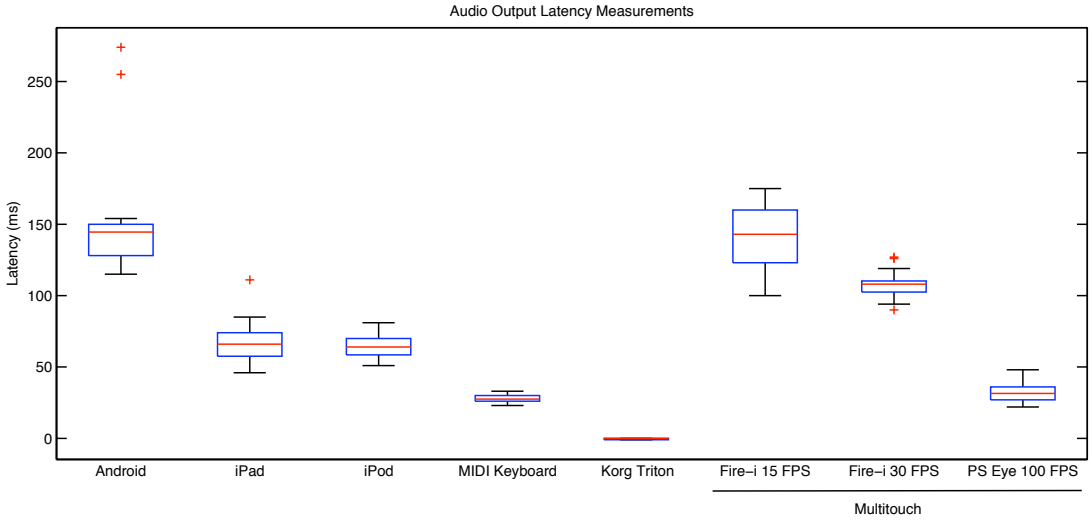


Figure 7: Results of audio latency tests.

4.4 Future Applications

This multi-touch table with haptic feedback has the potential to be deployed in educational environments, in addition to its current use as a software-synthesis performance controller. We are beginning to explore ways in which haptic-multi-touch interfaces can lead to engaging, fun, and collaborative music-making for children, in both schools and museums. We are also beginning work on an interactive soundscape-exploration system in which geographical maps, satellite images, and multi-channel soundscape recordings can be quickly navigated, explored, compared.

5. CONCLUSIONS

Our goal was to produce an economical multi-touch table with haptic feedback that exhibited low enough latency to be useful as a musical instrument. Our system succeeded in reducing average latency to 30 ms. There still seems to be some debate as to the exact JND for multimodal feedback, but subjective reports indicate that the inclusion of haptic feedback can improve the perception of simultaneity in new musical interfaces. This paper demonstrates proof-of-concept of the multi-touch table as a low-cost computer-music instrument with haptic feedback. We are creating a suite of software instruments for the table that we hope will leverage the multi-touch control paradigm with new tools for musical performance and composition, as the integration of multi-touch technology with haptic feedback provides many opportunities for creative exploration.

6. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant No. IIS 0757552 and by grants from the University of Miami. Thank you also to Pat O'Keefe and Mark Freeman for their work on our first multi-touch table.

7. REFERENCES

[1] B. Adelstein, D. Begault, M. Anderson, and E. Wenzel. Sensitivity to haptic-audio asynchrony. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 73–76. ACM, 2003.

[2] O. Bau, I. Poupyrev, A. Israr, and C. Harrison. Teslatouch: electrovibration for touch surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 283–292. ACM, 2010.

[3] E. Berdahl, H. Steiner, and C. Oldham. Practical hardware and algorithms for creating haptic musical

instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME-2008), Genova, Italy*, 2008.

[4] D. DiFilippo and D. Pai. The AHI: An audio and haptic interface for contact interactions. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 149–158. ACM, 2000.

[5] M. Eid, M. Orozco, and A. El Saddik. A guided tour in haptic audio visual environments and applications. *International Journal of Advanced Media and Communication*, 1(3):265–297, 2007.

[6] C. Gunn, M. Hutchins, and M. Adcock. Combating latency in haptic collaborative virtual environments. *Presence: Teleoperators & Virtual Environments*, 14(3):313–328, 2005.

[7] J. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM, 2005.

[8] V. Harrar and L. Harris. The effect of exposure to asynchronous audio, visual, and tactile stimulus combinations on the perception of simultaneity. *Experimental Brain Research*, 186(4):517–524, 2008.

[9] C. Harrison and S. Hudson. Providing dynamically changeable physical buttons on a visual display. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 299–308. ACM, 2009.

[10] G. Huang, D. Metaxas, and M. Govindaraj. Feel the fabric: an audio-haptic interface. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 52–61. Eurographics Association, 2003.

[11] Y. Jansen. Mudpad: Fluid haptics for multitouch surfaces. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, CHI EA '10, pages 4351–4356, New York, NY, USA, 2010. ACM.

[12] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, pages 139–146, New York, NY, USA, 2007. ACM.

[13] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. TUIO: A protocol for table-top tangible user interfaces. In *Proc. of the The 6th International Workshop on Gesture in Human-Computer Interaction and Simulation*. Citeseer, 2005.

[14] N. Lago and F. Kon. The quest for low latency. In *Proceedings of the International Computer Music Conference*, pages 33–36. Citeseer, 2004.

[15] D. Levitin, K. MacLean, M. Mathews, L. Chu, and E. Jensen. The perception of cross-modal simultaneity. *International Journal of Computing Anticipatory Systems*, 2000.

[16] T. Maki-Patola and P. Hamalainen. Effect of latency on playing accuracy of two gesture controlled continuous sound instruments without tactile feedback. In *Proc. Conf. on Digital Audio Effects, Naples, Italy*, 2004.

[17] T. Maki-Patola and P. Hamalainen. Latency tolerance for gesture controlled continuous sound instrument without tactile feedback. In *Proc. International Computer Music Conference (ICMC)*, pages 1–5. Citeseer, 2004.

[18] A. Mattek, M. Freeman, and E. Humphrey. Revisiting Cagean Composition Methodology with a Modern Computational Implementation. In *Proc. NIME 2010*, 2010.

[19] S. O'Modhrain and C. Chafe. Incorporating haptic feedback into interfaces for music applications. In *Proceedings of the International Symposium on Robotics with Applications, World Automation Conference*, 2000.

[20] K. Scherer and M. Zentner. Emotional effects of music: Production rules. *Music and emotion: Theory and research*, pages 361–392, 2001.

[21] J. Schöning, J. Hook, T. Bartindale, D. Schmidt, P. Oliver, F. Echtler, N. Motamedi, P. Brandl, and U. Zadow. Building Interactive Multi-touch Surfaces. *Tabletops-Horizontal Interactive Displays*, pages 27–49, 2010.

[22] A. Teiche, A. Rai, C. Yanc, C. Moore, D. Solms, G. Cetin, J. Riggio, N. Ramseyer, P. D'Intino, L. Muller, et al. Multi-touch technologies. *NUI Group*, 2009.

[23] L. Terrenghi, D. Kirk, H. Richter, S. Krämer, O. Hilliges, and A. Butz. Physical handles at the interactive surface: exploring tangibility and its benefits. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 138–145, New York, NY, USA, 2008. ACM.

[24] K. Walker, W. Martens, and S. Kim. Perception of Simultaneity and Detection of Asynchrony Between Audio and Structural Vibration in Multimodal Music Reproduction. In *Proceedings of the 120th Convention of the Audio Engineering Society, Paris, France*, 2006.

[25] E. Wenzel. Analysis of the role of update rate and system latency in interactive virtual acoustic environments. *AES Preprint*, 1997.

The Electromagnetically Sustained Rhodes Piano

Greg Shear
Media Arts & Technology (MAT)
University of California
Santa Barbara, CA 93106
gshear@mat.ucsb.edu

Matthew Wright
CREATE/MAT
University of California
Santa Barbara, CA 93106
matt@create.ucsb.edu

ABSTRACT

The Electromagnetically Sustained Rhodes Piano is an augmentation of the original instrument with additional control over the amplitude envelope of individual notes. This includes slow attacks and infinite sustain while preserving the familiar spectral qualities of this classic electromechanical piano. These additional parameters are controlled with aftertouch on the existing keyboard, extending standard piano technique. Two sustain methods were investigated, driving the actuator first with a pure sine wave, and second with the output signal of the sensor. A special isolation method effectively decouples the sensors from the actuators and tames unruly feedback in the high-gain signal path.

Keywords

Rhodes, keyboard, electromagnetic, sustain, augmented instrument, feedback, aftertouch

1. INTRODUCTION

The motivation behind this project comes from compositional experiments in the recording studio editing Rhodes piano samples in Pro Tools to create swelling and sustaining effects impossible to play on the original instrument. We desire these new affordances in a live performance setting controlled through the existing keyboard interface, extending standard piano technique all while leaving the original functionality of the instrument intact.

We present a novel system that offers limited control over the amplitude envelope of a Fender Rhodes electric piano, including infinite sustain, controlled by aftertouch on the existing keyboard interface. A primary design goal was to preserve the timbral qualities of the original electromechanical instrument, which rely on both the tone source (the vibrating tine) and the sensor (the magnetic pickup). With the addition of some circuitry and electromagnetic actuators to the existing electronics, we have extended the affordances of the instrument without compromising its original functionality.

2. EXISTING INSTRUMENTS

2.1 The EBow

The EBow is a device designed for sustaining vibrations in ferromagnetic guitar strings through positive feedback [2] and is controlled simply by moving the device toward or away from the strings. The EBow uses one sensing coil to generate a signal that drives a second coil which in turn exerts a time-varying magnetic force on the string supporting its oscillation. Permanent magnetic cores in each coil temporarily magnetize the ferromagnetic string greatly increasing efficiency of the actuator and allowing for both attractive and repulsive forces between the actuator and string. Without this magnetization, the actuator would exert only an attractive force on the string, effectively rectifying the actuator signal and adding undesirable high frequency distortion.

We found that direct magnetic coupling between the sensor and actuator coils leads to uncontrollable feedback in our system. There appears to be no compensation for this effect in the referenced EBow patent so we assume the EBow did not suffer from the same complications given the position and orientation of the coils and the amount of gain in the feedback circuit. Besides the compensation for this direct magnetic coupling, our electronics system is most similar to that of the EBow.

2.2 The Electromagnetically-Prepared Piano

The Electromagnetically-Prepared Piano [1] is an acoustic piano with electromagnetic actuators placed above certain strings. Each actuator is driven with an arbitrary audio signal (the creators suggest pure sine waves, orchestral samples, noise, etc.) through a standard audio amplifier and the strings filter the signal before acoustic amplification via the soundboard. Control is achieved through software such as Cycling 74's Max/MSP [8] and the original key/hammer action is left unaltered.

This differs from our system in that we drive the actuators with a signal generated by the vibrating mechanism thus completing a feedback loop. Furthermore, we control the system through pressure sensors retrofitted to the existing keyboard interface.

2.3 The Magnetic Resonator Piano

Andrew McPherson's Magnetic Resonator Piano [4] inspired our project and this is apparent in the similarity of our design goals. He also uses mechanical-electrical feedback to drive the piano strings but his actuator signals are generated through a much more complex system. A single piezoelectric sensor placed on the soundboard is the source for all of the actuators. This signal is distributed to a series of individually tuned bandpass filters that then drive phase-locked loops with adjustable delay to compensate for the propagation time through the soundboard. He achieves control through continuous sensing of each key with a modified Moog Piano Bar [5] and a complex mapping scheme of this control data to amplitude and spectral parameters for each note.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

3. THE FENDER RHODES

The Fender Rhodes piano [6] is an electromechanical instrument that uses a steel cantilever beam (the *tine*, seen in Figure 1) as its primary tone source. In each piano there is one tine per note with fundamental vibrating frequencies ranging from 27 Hz to 4.2 kHz on the 88-key model, and 41 Hz to 2.6 kHz on the 73-key model. Each tine is sensed by a dedicated passive magnetic pickup: vibration in the tine disturbs the magnetic field through a coil of wire thus generating an electrical signal. The average[1] of the signals from each sensor is present at the output jack of the instrument for amplification. Similar to an acoustic piano, the tine is struck by a hammer and damped by a felt pad. The *tuning spring* is a stiff wire wrapped around the free end of the tine that adds mass and allows for adjustment of the fundamental frequency.

3.1 The Tine and Tonebar

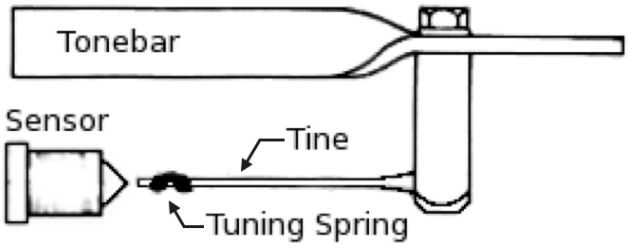


Figure 1: Tone generator assembly [6].

Each tine is paired with a *tonebar* and together they behave as an asymmetrical tuning fork. Although their fundamental frequencies are different, the tonebar stores energy from the initial hammer strike and helps to sustain vibrations in the tine [6].

Unlike a piano string, which is fixed at both ends and vibrates with overtones at near integer multiples of the fundamental, the tine is free at one end and has a decidedly inharmonic overtone series with the first overtone at a non-integer multiple several times higher than the fundamental (depending on the physical parameters of the tine) [7]. These inharmonic overtones give the Rhodes piano a somewhat bell-like timbre.

The tine itself is cylindrical (except near the base) with a diameter of 1.5 mm and lengths ranging from 18 mm to 157 mm. The free end of the tine swings up and down reaching a displacement of up to 50 mm for the longest tine, while shortest tine reaches a displacement of less than 1 mm.

3.2 The Pickup

The sensor (pickup) and vibrating tine behave nonlinearly, adding harmonic distortion to the sensor signal. The spectrum changes with their orientation: as the equilibrium point of the tine approaches the sensor axis, the fundamental and all odd harmonics are reduced, leaving the second harmonic as the strongest frequency in the series. Vertical adjustment of the tine (in the direction of oscillation) is known as *voicing* and the effect is consistent with the findings in [3] where a modeled guitar string oscillates perpendicularly to the axis of its pickup (motion similar to that of our vibrating tine with respect to the sensor). Figure 2 compares the spectra of two different tine alignments - one on the sensor axis (as seen in Figure 1) and the other 5 mm above the axis.

[1]Given all passive electrical components.

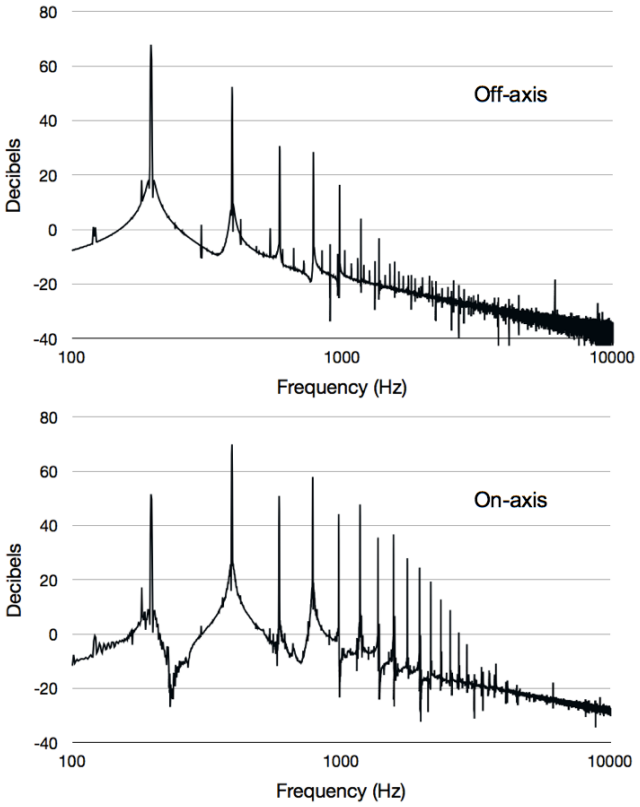


Figure 2: Variation in harmonic distortion due to sensor/tine alignment. Fundamental at 196 Hz.

4. ACTUATION

Driving the tine with electromagnetic actuators is straightforward given the large body of prior art, but in our case the magnetic pickups sense the driving magnetic field in addition to the tine thus directly coupling the actuation system with the sensing system. There are no obvious alternatives for either the sensor or the actuator: a piezo element in direct contact with the tine would change its resonant properties, and optical sensors are prohibitively expensive and would not add the same harmonic distortion described in Section 3.2. With these constraints we investigated two methods, driving the actuator first with a pure sine wave, and then with the signal generated by the sensor thus creating a feedback loop.

4.1 Synthesized Sine Wave

Driving the actuator with a pure sine wave at the tine's fundamental frequency initiated and sustained oscillations, but this strong driving signal completely dominated the signal generated by the tine as the pickup is sensitive to both. The actuator-sensor signal path introduces a scaling factor and phase shift that varies with frequency. This can be compensated for at a single frequency with a relatively simple circuit allowing us to subtract the pure sine wave and isolate the tine signal. Unlike filtering, this will not affect the signal when the actuator is inactive.

4.2 Feedback

We assume that actuation with a pure sine wave at the tine's fundamental frequency will not excite any of the non-harmonic overtones described in Section 3, whereas the mechanical hammer introduces energy over a wide range of frequencies exciting many of these overtones in addition to the fundamental. Once the tine has been struck, these overtones

Figure 3: Sustainer circuit with feedback.

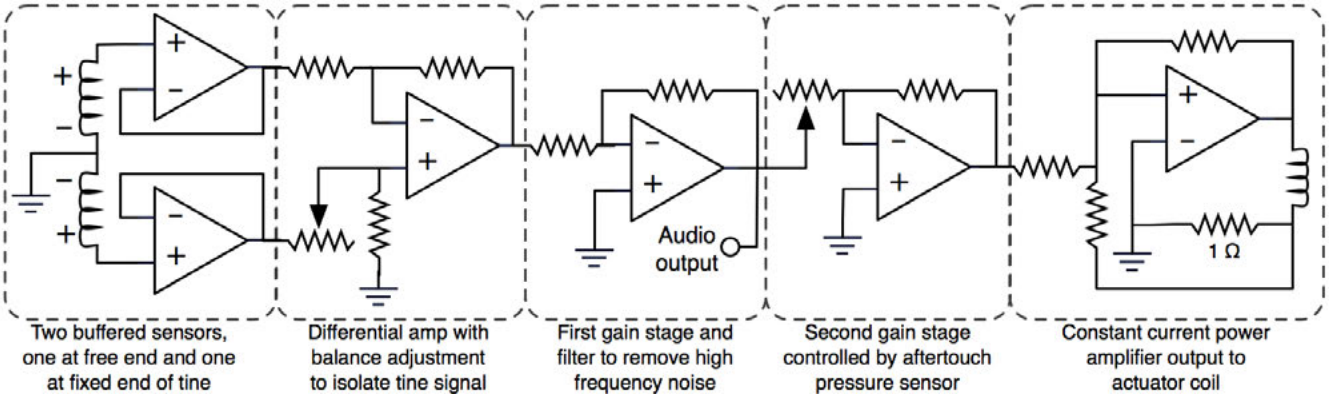


Table 1: Variable key.

Velocity of tine	v
Electromotive force	$\mathcal{E}$
Voltage across sensor, output amp	V_s,V_o
Current through actuator	I_a
Complex impedance of output stage	Z_{Out}
Circuit gain	G
Phase shift	ϕ
Magnetic field produced by sensor, actuator	B_s,B_a
Magnetic flux through sensor coil	Φ_B
Magnetic moment of tine	m
Force on tine	F
Distance from actuator of point on axis	x
Number of turns of wire in sensor, actuator	N_s,N_a
Cross-sectional area of actuator	A

should be present in the output signal and will self-sustain in the feedback loop.

4.2.1 Theory and Basic Equations For Control

We assume the vibrating tine is a damped harmonic oscillator that experiences a damping force proportional to its velocity v. To compensate for this and sustain oscillations indefinitely, the actuator must exert a force on the tine proportional to $-v$. The following equations (with variables defined in Table 1) show how the feedback system achieves this goal.

$$\frac{d}{dt}\Phi_B \propto v \cdot \nabla B_s \tag{1}$$

$$V_s = \mathcal{E} = -N_s \frac{d}{dt}\Phi_B \tag{2}$$

$$V_o = GV_s \tag{3}$$

$$I_a = \frac{V_o}{|Z_{Out}|e^{j\phi}} \tag{4}$$

$$B \propto \frac{N_a I_a A^2}{2(x^2 + A^2)^{\frac{3}{2}}} \tag{5}$$

$$F = \nabla(m \cdot B_a) \tag{6}$$

Equation (1) represents the relationship of magnetic flux rate of change to velocity of the tine traveling through the non-uniform magnetic field imposed by the sensor core. Equation (2) is the special case of Faraday's law for the EMF produced in a coil of wire. This also equals the voltage presented at the op-amp input assuming infinite input impedance. Equation (3) shows the voltage gain through the circuit. Equation (4) shows the phase relationship ϕ between actuator current and voltage. Equation (5) shows the magnetic field produced by a current through a coil of wire, simplified for the different magnetic permeabilities of the core and the air gap between the actuator and the tine. Equation (6) shows the force on the tine due to the magnetic field produced by the actuator.

The phase shift ϕ introduced at the output stage reduces actuator efficiency, and with a shift of more than 90° the actuator begins to damp the tine. A constant current amplifier (seen in Figure 3) is used to minimize this phase shift and Figure 4 shows the phase response curve. The theoretical calculations ignore the amplifier's output impedance and this may account for some of the discrepancy with the experimental data.

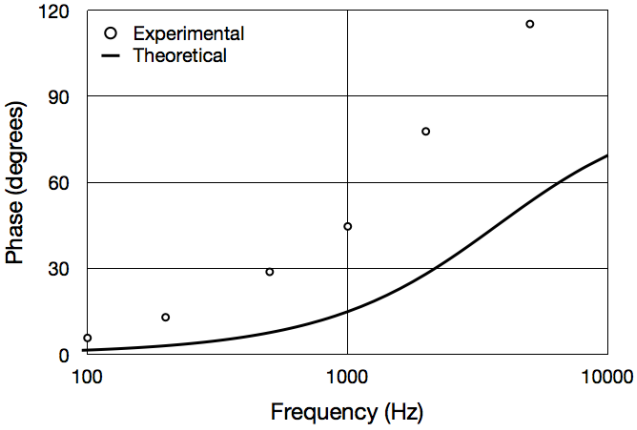


Figure 4: Phase response of constant current output amplifier.

4.2.2 Implementation

Again, direct magnetic coupling complicated early experiments as the high electrical signal gain far exceeded the attenuation of the magnetic field due to physical separation between actuator and sensor. Subtracting the actuator signal out of the sensor signal was necessary to control feedback; a second sensor with similar phase response placed near the fixed end of the tine (Figure 5) was used to provide the subtraction signal. In this configuration, the movement of the tine is detected by only one sensor, but the driving magnetic field is present at both sensors. Taking the difference of the two signals substantially removes

the actuator component and isolates the tine component. Please note that the distortion described in Section 3.2 is an effect of the physical vibration of the tine with respect to the stationary sensor; therefore, since the actuator is also stationary, no such distortion is imposed on the magnetic signal emitted by the actuator and received by the sensor.

Both sensors are original Rhodes piano pickups. The actuator is approximately 600 turns of 30 AWG copper wire wound around a plastic sewing machine bobbin mounted on a steel core. DC resistance is about 170Ω for the sensors and 11Ω for the actuator.

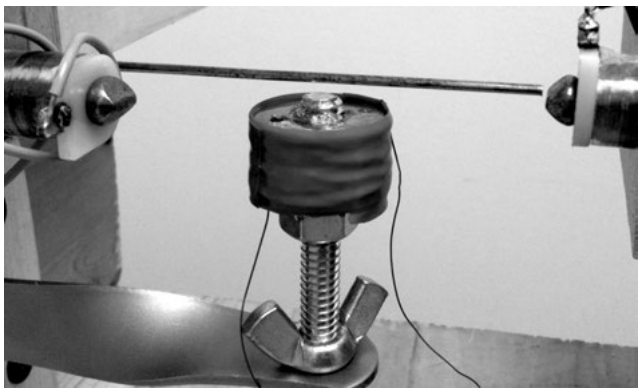


Figure 5: Actuator, two sensors, and tine.

5. PHYSICAL INTERFACE

Straightworward aftertouch control is achieved with a pressure sensor (variable resistor) placed on the keybed. This sensor has a resistance inversely related to applied pressure and is the input resistor on the second gain stage in the feedback circuit (Figure 3). This configuration maps aftertouch pressure to the rate of gain increase, within certain limits. Indeed, high pressure will quickly increase the signal through feedback to where the output amplifier clips severely and distorts. Decay time can be prolonged, but because our system (currently) lacks active damping the lower limit is governed by the natural decay of the tine.

6. RESULTS AND FUTURE WORK

Considering its simplicity, the control system is surprisingly effective. A wide range of arbitrary amplitude envelopes can be performed, including a slow attack achieved by exciting the tine with amplified noise in the system. Removing pressure against the keybed while still holding the damper away from the tine turns off the actuator and allows the note to decay naturally. See Figure 6 for a few examples of amplitude envelopes performed with this system. Subjective listening tests are also favorable - the perceived spectral quality of the electronically sustained note is the same as the naturally decaying note, though it is difficult to hear the difference between the two sustain methods. Driving the tine with a pure sine wave achieves reasonable sustain with a simple system, though we suspect the more complicated feedback method will be necessary if active damping is desired.

The actuator efficiency depends on the harmonic series of each note described in Section 3.2. We observe significant reduction in efficiency as the tine's equilibrium is adjusted towards the sensor axis and the second harmonic becomes the prominent frequency. Again, this voicing adjustment is important part of the instrument but we currently have no solution to the problem.

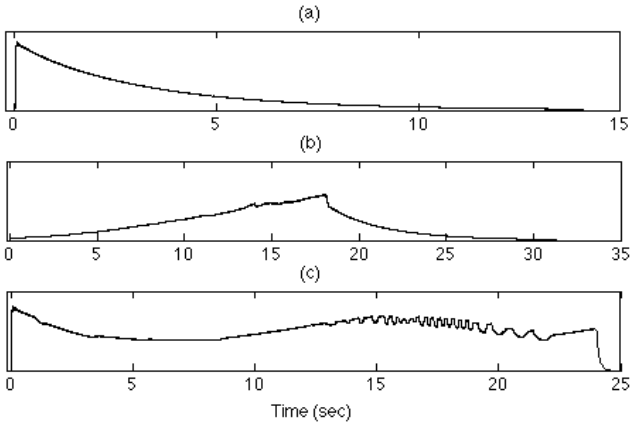


Figure 6: Amplitude envelopes of several notes produced by our instrument: (a) is a standard hammer attack with natural, unsustained decay; (b) also decays naturally, but reaches peak amplitude only by electromagnetic actuation; (c) is a standard hammer attack followed by tremolo and shortened decay by the felt damper.

Finally, the pressure sensor in the second gain stage (Figure 3) unsurprisingly adds a lot of noise to the signal path. Here a FET variable resistance would protect the signal while the pressure sensor provides a filtered control voltage.

7. ACKNOWLEDGEMENTS

We would like to thank Les Schaffer for his encouragement and patience while explaining the physics involved in this project, and Edgar Berdahl for his insights and ideas when we were coming up short on both.

8. REFERENCES

[1] E. Berdahl, S. Backer, and J. Smith. If I had a hammer: Design and theory of an electromagnetically-prepared piano. In *ICMC Proceedings*, 2005.

[2] G. Heet. String instrument vibration and sustainer, 1978. U.S. Pat. 4,075,921.

[3] N. Horton and T. Moore. Modeling the magnetic pickup of an electric guitar. *American Journal of Physics*, 77:144, 2009.

[4] A. McPherson and Y. Kim. Augmenting the acoustic piano with electromagnetic string actuation and continuous key position sensing. In *NIME Proceedings*, 2010.

[5] PianoBar. Products of interest. *Computer Music Journal*, 29(1):104–113, 2005.

[6] Rhodes Keyboard Instruments USA. *Rhodes Service Manual*, 1979.

[7] S. Whitney. Vibrations of cantilever beams: Deflection, frequency, and research uses, 1999.

[8] D. Zicarelli. An extensible real-time signal processing environment for Max. In *International Computer Music Conference*, pages 463–466, Ann Arbor, Michigan, 1998. International Computer Music Association.

Gamelan Elektrika: An Electronic Balinese Gamelan

Laurel S. Pardue
Responsive Environements
MIT Media Lab
75 Amherst St E14-548
Cambridge, MA 02142
punk@mit.edu

Andrew Boch
321 Highland Ave
Sommerville, MA 02144

Matt Boch
Harmonix
625 Mass. Ave, 2nd Fl.
Cambridge, MA 02139

Christine Southworth
65 Turning Mill Rd.
Lexington, MA 02420
southsea@kotekan.com

Alex Rigopulos*
Harmonix
625 Mass. Ave, 2nd Fl.
Cambridge, MA 02139

ABSTRACT

This paper describes the motivation and construction of Gamelan Elektrika, a new electronic gamelan modeled after a Balinese Gong Kebyar. The first of its kind, Elektrika consists of seven instruments acting as MIDI controllers accompanied by traditional percussion and played by 11 or more performers following Balinese performance practice. Three main percussive instrument designs were executed using a combination of force sensitive resistors, piezos, and capacitive sensing. While the instrument interfaces are designed to play interchangeably with the original, the sound and travel possiblilities they enable are tremendous. MIDI enables a massive new sound palette with new scales beyond the quirky traditional tuning and non-traditional sounds. It also allows simplified transcription for an aurally taught tradition. Significantly, it reduces the transportation challenges of a previously large and heavy ensemble, creating opportunities for wider audiences to experience Gong Kebyar's enchanting sound. True to the spirit of oneness in Balinese music, as one of the first large all-MIDI ensembles, Elek Trika challenges performers to trust silent instruments and develop an understanding of highly intricate and interlocking music not through the sound of the individual, but through the sound of the whole.

Keywords

bali, gamelan, musical instrument design, MIDI ensemble

1. INTRODUCTION

Gamelan has been performed for hundreds of years in Indonesia. The term gamelan is a general reference to a musical ensemble which can take many forms. One of the most famous is the metalophone instruments of the Balinese Gong Kebyar. It is renowned for the shimmer, intricate elaborate melodies and the tight interlock and togetherness of the playing ensemble. Uniquely in Balinese gamelan, the instruments come in pairs, where each instrument is slightly out of tune with the other half of the pair resulting in acoustical beats. A characteristic of Balinese composition is the interlocking of parts; a single line is regularly split between two instruments and two players resulting in quick, intricate rhythms. Additionally, gamelan is based on different versions of pentatonic tuning with each gamelan set having its own related but distinct tuning. No two gamelans are the same [3].

Figure 1: Galak Tika's *gangsa* and *reongs* including instruments from the Beta gamelan at the rear.

Balinese Gamelan is immensely popular in Bali, which hosts large national ensemble competitions. Study of the instruments first spread to the US in 1958 [8]. Balinese works are through composed and taught aurally. The Massachusetts Institute of Technology hosts Gamelan Galak Tika (GGT), founded in 1993 under the direction of Evan Ziporyn. With regular performances around the East Coast including Carnegie Hall, Lincoln Center, Brooklyn Academy of Music, and the Bang on A Can Marathons, GGT is a musical innovator focusing on new works by both Balinese and American composers. GGT owns two sets of instruments. The first set was somewhat hard to blend to Western tonalities so a second set, the Beta gamelan, was commissioned from Bali with a different tuning more suitable for projects with Western instruments. Tunning is not just a problem for GGT. Dewa Ketut Alit, one of the foremost composers in Bali, has also turned to working with multiple gamelans for a larger pitch range within a single composition.

Tuning is not the only limitation gamelan faces. The en-

*The company Harmonix was not involved in this project.

semble itself is large and very heavy. The instruments are metalaphones, including large gongs, brass or bronze pots and *gangsa* with 1/2 inch thick brass/bronze keys and solid heavy frames. Transporting the gamelan to a performance is a significant task on its own. With the mundane issues of tuning and transport continually complicating commissions and performance, through the generous support of Alex Rigopulos and Sachi Sato, GGT's Gamelan Elektrika was born: a more compact electronic version of the original instruments. Since the new instruments provide MIDI, Elektrika simultaneously gave access to new sounds and new tunings with as close to the original interface as possible. Another benefit is that MIDI is easily transcribable, proving a valuable tool to any Balinese ethnomusicologist documenting the aural tradition.

Gamelan Elektrika is not just about the instruments, but as gamelan should be, it is about the ensemble. Elektrika involves at least 11 performers playing often highly detailed interlocking parts on silent MIDI instruments. Any serious musician knows how important audible feedback from their instrument is, yet Elektrika is an essentially silent orchestra. MIDI signals are routed to a single brain that has complete control over the player's sound. We believe this is the first significant large ensemble of this form.

2. THE GAMELAN INSTRUMENTS

Gamelan Elektrika is a subset of a *gong kebyar* ensemble. *Gong kebyar* is presently the most popular form of gamelan within Bali and is usually the focus for the most ambitious compositions.

A full *Gong Kebyar* can have 24 or more instruments tuned to a pentatonic scale termed *pelog*. There are four instruments that are almost exclusively percussion: the *ceng ceng* playing rythmic ornamentation, the beat-keaping *kempli*, and a pair of hand drums called *kendang*. Four large gongs punctuate phrase structure. A single stringed bowed instrument, the *rebab*, and the *suling*, a flute, provide melodic ornamentation. Along with the gongs, the instruments we built were the main melodic instruments of the ensemble. These main melodic instruments include the *reong*, a set of kettle pots, and both the *pokok* and the *gangsa* which are similar keyed metal xylophones hit with mallets called *pangguls* [9].

2.1 The Pokok

The *pokok* are the melodic core played with rubber tipped wooden *pangguls*.

1) Jegogans- This pair is the lowest pitched set beside the gongs and covers the 5 tone pentatonic octave. It generally outlines key melody notes.

2) Jublags- The next pair up in range with slightly faster notes and playing a more complete subset of the melody. The jublags have 5 or 7 keys.

3) Penyacah- A pair of 7 keyed instruments above the jublags. These generally play along with the primary *ugal* melody.

2.2 The Gangsa

The *gangsa* provide melodic elaboration and are played with hard tipped wooden *pangguls*.

4) Pemade- Two pairs of two octave 10 keyed instrument that provide the mid-range.

5) Kantils- The highest instruments in the gamelan, harmonizing with the *pemade*. There are two pairs of these

6) Ugal- Also spanning two octaves with 10 keys, it is the only unpaired *gangsa*, generally linking with the *kendang* to lead the group. It plays the primary melody.

2.3 The Reong

Also spelled reyong, it plays melodic and rhythmic elaboration and is played with wooden sticks wrapped in string.

7) The Reong- Consists of 12 tuned kettle pots spanning just over two octaves. These can all sit on one frame or be split between two. Considered one instrument, it is played by 4 people with the higher octave usually doubling the lower. It is played similarly to bell ringing where an individual is responsible for only specific pitches within the melody.

2.4 Ensemble Play

Two major structural components of the music style are *kebyars*- very fluid unmetered and variable interruptions (literally to burst open) and *kotekans*-tight interlocking sections where the melody is formed by the combination of two separate parts [3]. The *gangsa* and *reong* are the primary *kotekan* instruments. For *kotekan*, players are paired rather than instruments with half the *gangsa* (excluding *ugal*) playing *polos* and half playing *sangsih*. Although the interlock is rarely as straight forward as single note-to-note on-beat, off-beat interlock, *polos* primarily centers around the beats with *sangsih* filling in the off-beat. Outside *kotekans*, *sangsih* usually plays the melody in a range above the *polos*. The *reong* often plays in *kotekan* with itself or plays unison rhythmic punctuations matching the rhythm section [9].

3. RELATED WORK

Gamelan has recieved little engineering attention. Aaron Taylor Kuffner, and Eric Singer, with LEMUR built the Gamelatron, a robotic gamelan orchestra [2]. The Gamelatron is a traditional set of instruments played robotically rather than a gamelan interface. Alternatively, Ajay Kapur has looked at custom percussion controllers for traditional instruments with the EDholak and ETabla [5]. These are Indian instruments but involve similar approaches applied towards different ends.

Meanwhile, electronic MIDI interfaces replicating traditional instruments are hardly a new thing. The first MIDI interfaces were keyboards produced in 1983 by Roland and Sequential Circuits. Originally intended as mere controllers, they were a clear mimic of the piano and quickly evolved to fully replicate the it's full range of interaction. 1984 saw Roland release the G707, a MIDI guitar controller followed by a MIDI only drum pad, the PAD8 'Octapad', in 1985 [7]. The flexibility of MIDI means it has been found commonly in new instrument interfaces ever since.

Gong Kebyar instruments are closest to xylophones and vibraphones in performance style although the damping techniques of both the *gangsa* and the *reong* are unique. The sustained ring of the natural instruments and the playing technique developed around it means damping is much more integral to the instruments. Alternate Mode's malletKat is the closest MIDI xylophone to fit the capabilities of gamelan. It uses FSRs (force sensitive resistors) to detect note onset and damping. It is also compatible with any mallet, matching a design goal to retain traditional playing feel including real *pangguls* [1]. Wernick uses piezos in its XyloSynth and is not mallet specific, but damping is based on time release [4]. Don Buchla's Marimba Lumina uses a very different approach based on radio frequency technology which enables capture of more playing style but requires special mallets [6]. None of these instruments were designed with gamelan damping techniques in mind and hence, none of them were sufficient for Elektrika.

4. ELECTRONIC GAMELAN DESIGN

Although Elektrika is intended for long term use, initial design and construction was for the premiere of Christine Southworth's "Super Collider" with Kronos Quartet at the Lincoln Center Aug 13, 2010. Part of Kronos's commission of the work was that it would be performed with electronic gamelan and, to improve touring viability, it should be a smaller ensemble. Hence rather than the full 24 instruments, the ensemble was reduced to 13 instruments: 4 *gangsa*, 2 *pokok*, 4 gongs, and a traditional rhythm section but with only one *kendang*.

Traditionally the *gangsa* and *pokok* parts are doubled. The instruments are paired and slightly out of tune with each other creating Balinese music's shimmering quality. Since the instruments are sampled, the idea is that one player can now trigger both pitches. This let us reduce the metalaphones to the minimum players.

The *gangsas* remain split between *pemade* and *kantil* with a *polos* and *sangsih* player for each section while the *pokok* is reduced to one *jublag* and one *jegogan*. *Penyacah* is often optional in traditional Gong Kebyar and was left out. There is also no *ugal*. There is no substitute for a full *reong*: two frames house 12 synthetic pots. The gongs, previously the most massive and heavy part of the ensemble, have been moved onto one significantly more portable frame.

The rhythm instruments have remained the traditional acoustic versions. There were never any plans to build a new hand drum interface for the *kendang* as commercial electronic drum systems are readily available. After testing a couple, they were not to taste, so we stayed with the acoustic originals. There were plans to build a *kempli* and, as the instrument is similar to a *reong* pot, it is not a significant technical challenge. However the *kempli* keeps the beat and is what players lock onto. It is preferential for its sound to remain centered within the ensemble and while slight latency or problems with another instrument can be dealt with, latency or missed notes on the *kempli* could be disastrous. A guaranteed anchor becomes especially important as the other instruments do not actually make noise. The *ceng ceng*, although desired, has yet to be built due to time considerations. Being a small ancillary instrument requiring a unique engineering solution, leaving it acoustic for "Super Collider" and augmenting it with a drum pad for the few synthetic sections was deemed acceptable.

There were a few dominant themes in the design goals for the instruments. As previously mentioned, the instruments should be lighter and more compact for travel. They should also retain as much of their original feel and performance technique as possible. Being able to use the normal mallets was preferable but not required. Meanwhile, the overall ensemble performance included the plan for instruments that could change samples and effects on the fly during performance but have to remain intuitive and understandable enough that the performers can still meet the demands of coordinating complex parts with the other performers.

4.1 Gangsa and Pokok Design

Focusing on the *gangsa* requires further discussion of playing technique. Proper playing of the *gangsa* (and *pokok*) involves striking the key with a *panggul* in the right hand while damping the previous note by grabbing the end of that key with the left hand. This creates the effect of one hand trailing the other. Musical texture can be varied by changing how long a struck note is allowed to ring and use of a "closed" hit meaning the key is damped while struck.

The *gangsa* have gone through two major design iterations. As fingers are used for damping, the initial design idea used in the first performance was to use a piezo to detect strike and strike velocity and a touch capacitive sensor to detect damping. First prototypes were made using acrylic but transitioned to cast urethane rubber keys. This was done as the rubber acts as a good acoustic dampener when hit by hard wood mallets and also has sufficient spring for good recoil. Casting also allowed the electronic keys to physically mimic the originals.

We used Vytaflex 20 for it's bounce and color, backing it with a 1/4 inch acrylic sheet. A large 1 inch piezo disc was centered on each piece of acrylic before casting and a copper plate added at one end. The piezo was sandwiched between the acrylic and the urethane while the copper capacitive plate resided underneath exposed to touch.

Figure 2: *Pemade* with rubber tipped *panggul*. Two FSRs between acrylic detect central strike and edge damping.

An Arduino Mega, able to support the analog inputs from upto 10 keys, was used to process the signals generated on the *gangsa*. Aside from light conditioning, the piezo signals from each key went directly to the chip analog inputs where they were polled. The capacitive signals were processed first using a single Atmel QT 1103 capacitive touch sensor which subsequently sent the digitized results to the Arduino.

Integrating the size of the piezo strike provided velocity. *Gangsa* and *pokok* are played one note at a time which also made cross-talk largely a non-issue. The strongest fastest signal is always the target signal. The capacitive sensing design was more challenging as the pad sizes were quite large (starting from 1 inch x 1 inch) and had to be tuned. When too sensitive, passing the hand near the sensor, as is a common in performance, triggered unintentional damping.

During construction and testing of the first gangsa design, we found that inconsistency in the casting thickness and piezo placing and adhesion, meant that velocity response was insufficently uniform and non-intuitive. Working with piezo discs, the adhesive and mounting significantly impact the quality of signal recieved so slight differences lead to comparative inconsistency. For an instrument made by hand, sufficient consistency is hard to achieve. Adding to this, the urethane damps too effectively. The physical hit does not propagate adequately throughout the whole key meaning that a hit far from the piezo registers more weakly than a hit directly above the piezo even though the physical force used was the same. A simple calibration test was devised using a ping pong ball dropped down a paper towel roll. Although detecting the strike was reliable and repeatable, the velocity sensitivity was not and deemed insufficient for use in performance.

These issues were addressed with a significant redesign after the first performance. To fix issues with velocity consistency, each key on a *gangsa* now uses an FSR sandwiched

between acrylic with a second FSR to detect damping from a hand squeezing. Thin foam spaces the acrylic appropriately to the sensor size. FSRs can be slow to decompress and return to original state, an issue handled by use of a moving baseline. After considering playing styles, it was decided that the damping rate of the real instrument directly corresponds to the pressure with which it is squeezed and damping would be more appropriately detected through pressure. Conveniently, switching away from capacitive sensing resulted in more reliable damping, although frequent calibration was required while the new instruments settled. With the struck surface now hard acrylic, accoustic damping of the strike is achieved by attaching urethane to the *panggul* tip. Although a slight divergence from the intended goal to use unmodified *pangguls*, it is a minor modification not significantly changing feel.

The *pokok*, being similar in playing technique to the *gangsa*, have used the same designs adapted for correct scale.

4.2 Reong Design

The 12 pots of a real *reong* rest on strings providing bounce and enabling resonance. It has four players who each have two string wrapped *pangguls* and play single note melodies or chords. There are two primary styles of hit, the *byong* and the *chuck*. The *byong* is produced by striking the boss or nipple of the pot with the string wrapped part of the *panggul* and produces a clear pitched tone while the *chuck* is produced by hitting the flat part of the pot below the boss with the *panggul*'s hard wooden tip. The *chuck* is less tonal and more percussive.

Like the *gangsa*, the *reong* pots ring significantly and are damped for musical clarity and texture. Damping is achieved by direct pressure applied using the *panggul* with a technique of double hitting. The first strike is allowed to ring while the second, quieter damping strike sustains enough pressure to mute the original. In practice this is done very quickly and is hard to master. The *reong* also features a closed hit which is one that is never allowed to ring. Both *byongs* and *chucks* can be damped this way although the decay from a *chuck* is fast enough that damping is not as significant a concern.

The physical design of a *reong* pot is an acrylic mimic. A solid acrylic column topped with soft rounded Vytaflex urethane rubber is used for the boss and is mounted freely in a removable acrylic pot. A special rubber *chuck* pad sits in the pot edge. The *byong* column is kept in place by a base it slots into. Felt is placed between the column and the pot to eliminate contact sounds and ensure proper fit.

Byong strikes are measured using a piezo film placed within the base the column slots into. This enables the piezo to stay in a stable location. An FSR is co-located which is used to detect pressure and damping. The piezo response occurs faster than the FSR response and is easily tuned to capture a full range of velocities through signal integration. Using the FSR alone was not done as the column is tightly centered by the felt and the rest position and pressure from the *byong* column are not always consistent. Original tests also showed issues with dynamic range and the slow response of the FSR increased latency. The combination of the FSR and piezo has also proven very handy for identifying cross-talk. The *byong* sensors are sandwiched with foam and felt to provide some isolation from vibration transmitted through the frame and the column. Lastly, after the debut performance, it was found that the addition of a light weight disc spring isolates the column from the sensors, dramatically reducing cross-talk signals for much improved low-level sensitivity.

The *chucks* were originally built with a piezo film beneath a urethane pad combined with two FSRs in the pots rubber legs. The electronics will actually support four FSRs which could be used for position sensing but only two have been used so far. These sense pressure on the pot indicating the pot is damped.

During construction it turned out that the sensors are fairly delicate and would break easily while placed in the pot legs. Additionally, as with the *gangsa*, the urethane did not transmit the strike evenly so that the *chuck* velocity was strongly linked with where and how it was hit. After the debut performance, the piezo was removed and the two FSRs were moved to directly beneath the pad. Being more directly pressure sensitive, the FSRs remain largely cross-talk impervious. Using two FSRs enabled coverage for the full pad area and securing them under the pad is a less risky location for breakage.

Figure 3: One half of the *reong* holding six pots. The center is a free-standing rod with piezo films and FSRs placed underneath used to implement *byongs*. The small pads on the right are the *chucks*.

With up to seven inputs per pot, each *reong* pot has its own sensing engine. The piezo and FSR signals are run through operational amplifiers for gain control and maximum dynamic range before being analyzed using an Atmel Mega 88. Although initially interrupt driven, polling turned out easier and sufficiently effective. The latency for sending is under 7 ms with a 70 ms debounce hold after which it begins checking for damping.

Unlike the *gangsa* where cross-talk is easily ignored by selecting the peak signal, up to four of the six pots on a *reong* half can be played simultaneously. Additionally, a *reong* pot is uninformed of a neighbor's signaling and the sensors are affixed to the frame recieving signal propagation through it. The paired behavior of the FSR and the piezo enable the differentiation between cross-talk and a hit. With both the *byong* and the *chuck* in the case of cross-talk, the piezo response is timed significantly different from the FSR response. The column and the pot both being free standing, the FSR does suffer from cross-talk. With the *byong* column, the FSR signal is caused more from the column landing after a cross-talk induced disturbance so that it significantly lags the piezo response to the initial vibrations. The FSR signal also often rises before dropping as the column bounces.

Although each pot has its own MIDI out, for efficient cabling, each half of the *reong* also uses a Parallax Propeller

to combine the MIDI streams on a frame for a single out. The Propeller, with eight parallel cores, turned out to be perfect for this task as a core could be devoted to tracking each of the six MIDI inputs and queing input messages in a stack to be sent out through an output core. This alleviated any concerns for collisions and means if the maximum of four pots are hit simultaneously, it adds a maximum of only 5 ms latency and guarantees no missed messages. The MIDI combining board also acts as a power distribution source for the other boards.

Figure 4: Gamelan Elektrika gongs being performed by Mark Stewart and Jacques Weissgerber. Rear mounted piezos discs are used to detect the strike while centered gold discs provide damping through capacitive sensing. *Photo by Kevin Yatarola*

4.3 Gong Design

After toying with a couple of novel designs it was decided to use the same sensing arrangement as the first *gangsas* for the gongs. Each of the four gongs have a piezo attached to a large acrylic disk to detect when it has been hit. Damping (not musically required but immensely useful as gongs have extremely long resonance) is done by mounting a large copper vinyl circle at the center. This is also visually suggestive of a gong. Again, the electronics are simply a reduced version of the original *gangsas*. As the strike surface is acrylic, the gongs do not suffer the same strike propagation issues. The inital design and construction proved itself well in the first performance and has only needed minor maintenance.

Structurally, the size and weight of the gongs have been dramatically reduced. Four large stands have been combined into one that holds four free swinging acrylic disks. Their sizes range from 9 to 12 inches in diameter, functionally matched to the real gongs according to their size.

4.4 The Brain

According to performance tradition where a musician is in full control of his sound, each player would be able to monitor and select sound banks locally. However in our case, individual control of a sound bank would be both a physical and mental challenge as the hands are fully occupied during a sitting performance. It seems appropriately Balinese to consolidate sample bank control in a central brain. Also, individual monitoring though highly advantageous, ends up either a cabling and mixing nightmare due to scale. A local synthesizer module is presently prohibitively expensive.

The result was a risky but necessary decision to generate all sound through one computer. The brain is a Macbook running Ableton which a musician uses to select sample banks in real-time. MIDI input boxes take 10 different instrument inputs and pass the MIDI note information on to Ableton. This has a disconcerting effect on a musician due to the lack of direct feedback from the instrument, replaced with the need to pick "your" sound out of a full mix. Added to this, the sample bank may change without the musician's input. Performance in this environment requires trust in the instrument behavior, low latency, and a different level of physical comfort than normal with a piece. Technical challenges aside, it was not certain from a musical perspective whether this would be a feasible performance environment.

5. RESULTS

Southworth's piece "Super Collider" successfully debuted as planned to an audience of over 5000 on Aug 13, 2010 at the Lincoln Center performed by Kronos Quartet and Gamelan Galak Tika using Elektrika. Apart from some issues eventually traced to electrical interference in cable runs, the final instruments worked largely as intended though they lacked velocity sensitivity. The decision to drop velocity was as much due to musicians inexperience with the instruments as technical challenges.

Kronos Quartet rejoined for a second performance on April 15, 2011, at MIT's Kresge auditorium to a sold out crowd using the second version of the gangsas. This time there were no problems from the instruments. Improved instrument sensitivity and reliability plus rehearsal time enabled the return of instrument dynamics. Both performances were a success.

Rehearsals smoothed some of the disconcerting effects of sample bank changes and transpositions not triggered by the musician. The lack of local monitoring has displayed itself to be a challenge but is not insurmountable. An example of the trouble it can cause is there are times the *pemade polos* part is in unison with the *kantil polos*. If one player is slightly ahead or off, it becomes very difficult to know which player is actually the one ahead as there is no sound identification and correcting by trying to slow back a bit could just make the problem worse.

The ease of transistion from learning on the original instruments proves Elektrika's success at being interchangeable with the real instruments but dynamics remain a challenge. A musician learns an instrument through feedback-hitting with a particular force produces a particular level of sound. For Elektria, sound systems and outside sources also effect volume, so a definite understanding is hard to come by especially without the chance to learn in a solo environment. Due to set up and rehearsal constraints this is yet to be mastered.

The redesigns for the second performance have yielded robust and satisfactory instruments. The instruments play consistently and sufficiently similarly to the originals. The gongs never needed much revision. The move to FSRs for the *gangsa* provides more expression at the cost of slight latency due to FSRs slow response time. Interestingly, the latency is small enough that musicians cope very quickly with it once aural feedback is available. Each time a new sound setup is used with different monitoring paths, the audible latency changes slightly regardless of the instruments. This requires learning new timing. At each new setup, the melody starts with a swing as interlocked syncopations are slightly off. However the section adapts and evens out after just a few minutes of sectional practice.

The addition of the disc spring after the first performance largely eliminated sensitivity limitations for the *reong byongs*. It is responsive and reliable. Now the question is

Figure 5: Gamelan Elektrika played by Gamelan Galak Tika during its premiere with Kronos Quartet at the Lincoln Center. The *reongs* and two *pokok* can be seen on the far side with the four *gangsa* and gongs on the left. The rear center features the accoustic percussion section, the computer "brain", and traditional gongs which Kronos Quartet played as part of the performance. *Photo by Kevin Yatarola*

whether the damped sound can be achieved using the same sample as the open hit. The move to two FSRs for the *chuck* works and has resulted in fewer torn sensors than the first design, but is otherwise not a major improvement.

6. FUTURE WORK

With the first two performances complete, Gamelan Elektrika is next bound for the Bang On A Can Summer Festival. It will be used in the composition course to teach gamelan and available for use by the students. Terry Riley, previously reluctant to write for GGT, has also been commissioned to write a piece for performance in the spring of 2012.

The second rebuild of the gangsas has settled well at this point and is now plug-and-play with no forseeable significant changes. The reong is due to move to a new larger frame that can effectively protect the electronics, presently exposed underneath. The *byong* has proven reliable, sensitive, and expressive and will merely require being rebuilt in the new frame. Switching to just FSRs to sense *chucks* for the second performance has resulted in a confusing mixture of latencies as the piezo driven *byong* is much more instantaneously responsive. During the rebuild, the piezo will be re-incorporated for consistent response time. To mitigate the directionality of force applied to the urethane, the pad will also be given a firm base so pressure applied anywhere on the pad can be equally detected.

Long term technological goals are to provide an in-ear monitor capability at each instrument, or instrument pair, enabling the musician to easily hear and distinguish what they are playing. There are also plans to finish the as yet undesigned electronic *ceng ceng* and the option of the electronic *kempli* in order to complete all the electronic instruments originally envisioned.

7. CONCLUSIONS

The Gamelan Elektrika instruments work. They achieve both primary design goals- a significantly reduced physical size and weight and a similar feel to the original. Through MIDI we have been able to meet the musical goals of variable tunings, wider sound palette, and easier transcription, The initial success even on immature instruments has been brilliant, and gamelan can now participate in the electronic age of music that other instruments entered years ago. For performers and musicians in the genre it is a freeing and exciting development.

8. ACKNOWLEDGMENTS

Tremendous thanks to Alex Rigopulos and Sachi Sato for their funding, support, and guidance without which Elektrika would be just an idea. Thanks also to Quentin Kelley for building us beautiful frames, Galak Tika director Evan Ziporyn, Jaques Weissgerber, Katie Puckett, Julie Strand, Bill Tremblay, Noah Feehan and Steph Bou and the rest of the gamelan for their help, support, and patience. Thanks to Kronos Quartet for such a brilliant debut and lastly thanks for the support of Dr. Joe Paradiso, Responsive Environments and the MIT Media Lab.

9. REFERENCES

[1] http://www.alternatemode.com.

[2] http://www.gamelatron.com.

[3] L. Gold. *Music in Bali: experiencing music, expressing culture*. Oxford University Press, 2005.

[4] M. Goldstein. Playing Electronics with Mallets Extending the Gestural Possibilities. 2000.

[5] A. Kapur, P. Davidson, P. Cook, P. Driessen, and W. Schloss. Digitizing north indian performance. In *Proceedings of the International Computer Music Conference*. Citeseer, 2004.

[6] L. t. O. Nearfield Multimedia Marimba Lumina. *Electronic Musician*, 16(6), June 2000.

[7] G. Reid. The History Of Roland. Part 2: 1979-1985. *Sound On Sound*, December 2004.

[8] P. J. Revitt. The Institute of Ethnomusicology at U.C.L.A. *College Music Symposium*, 2, 1962.

[9] M. Tenzer. *Gamelan gong kebyar: the art of twentieth-century Balinese music*. University of Chicago Press, 2000.

Sonicstrument: A Musical Interface with Stereotypical Acoustic Transducers

Jeong-seob Lee and Woon Seung Yeo
Audio & Interactive Media Lab
Graduate School of Culture Technology, KAIST
335 Gwahangno, Yuseong-gu, Daejeon, Korea
jslee85@kaist.ac.kr, woony@kaist.edu

ABSTRACT
This paper introduces *Sonicstrument*, a sound-based interface that traces the user's hand motions. Sonicstrument utilizes stereotypical acoustic transducers (i.e., a pair of earphones and a microphone) for transmission and reception of acoustic signals whose frequencies are within the highest area of human hearing range that can rarely be perceived by most people. Being simpler in structure and easier to implement than typical ultrasonic motion detectors with special transducers, this system is robust and offers precise results without introducing any undesired sonic disturbance to users. We describe the design and implementation of Sonicstrument, evaluate its performance, and present two practical applications of the system in music and interactive performance.

Keywords
Stereotypical transducers, audible sound, Doppler effect, hand-free interface, musical instrument, interactive performance

1. INTRODUCTION
Sonicstrument is a simple but powerful interface that detects the user's hand motions with sound. The system does not utilize any ultrasonic sound and related devices; instead, it consists of a pair of stereotypical earphones and a microphone which transmit and receive signals whose frequencies range within the transducers' bandwidths (mostly covering human's theoretical audible range) but are barely perceptible for most people. To assure its reliability in an acoustically uncontrolled environment (e.g., loud ambient noise), the system performs Doppler analysis in signal processing to detect the transducers' motions in one dimension.

Sonicstrument aims to provide the simplest hand gesture interface for general users including interactive media artists. The system utilizes commonly used bud earphones and a microphone as transmitter and receiver, and does not incorporate any extra hardware such as special ultrasonic transducer and/or wireless system that may require technical expertise to use. Also, the small and handy nature of the earphones controlled by the user makes the system highly practical and suitable for interactive performance.

As a musical interface, sonicstrument has been featured in two different performances with contrasting scenarios: 1) a smaller-size, near-field environment for computer-synthesized virtual instrument performance, and 2) an interactive dance performance at a larger scale. In both cases, the system successfully traced the motion of the user.

This paper is organized as follows: we first discuss the detection mechanism of the Sonicstrument based on the review of previous studies, and describe the design and implementation of the system. Finally we present two application examples mentioned above.

2. RELATED PRIOR WORK

2.1 Motion Detection
Motion detection has been widely adapted to numerous studies in a variety of fields ranging from pure scientific research to practical fields (i.e., sports and security) and from music to media art.

Numerous motion detection systems have been developed with a variety of detection mechanisms and sensors. Examples include sound (acoustic transducers), optics (cameras, infrared sensors), electromagnetic field (compass), and motion/vibration (accelerometers) [7]. For short-range or indoor motion detection, ultrasonic signals – sounds with frequencies greater than the highest limit of human perception – are frequently selected due to the following:

- Although systems with radio-based location techniques such as the Global Positioning System (GPS) perform well in widely open areas, they are prone to severe multipath effects when used inside buildings.
- Electromagnetic sensors may be interfered with by unexpected magnetic fields as well as metal structures.
- Moreover, optical systems generally require expensive imaging detectors and suffer from line-of-sight problems [4].

While ultrasonic motion detection systems are free from these problems and are deemed suitable for moderate-scale indoor applications, they require special transducers, which can be expensive, and/or require technical expertise for use and implementation, thereby imposing practical limitations in terms of cost and technology.

2.2 Sound-based Motion Detection
Sound can be considered as a mechanical phenomenon that contains information about a physical "event." Many attempts have been made to detect, analyze, and classify certain events only from their sounds [3, 11 13].

In this paper, we focus on sonar-style examples that use sound "propagation" characteristics for detection, most of which are based on Time-of-flight (TOF) observation [4, 6, 9, 10, 12]. In other words, these sonar-style examples consist of mobile beacons and static receivers that communicate Radio Frequency (RF) and ultrasound signals. The RF channel triggers both beacons and receivers to transmit/detect a pulse, and the time delay between the RF trigger and detected sound – assuming constant speed of sound and negligible RF propagation time – becomes the TOF and is used to calculate the distance between a beacon and a receiver. This one-dimensional (1D) distance detection can be expanded to three-dimensional (3D) localization by deploying multiple receivers and using trilateration method.

TOF method suffers from limited precision due to the irregular time delay of the system process. To solve this problem, Lopes et al. suggested a localization method that compensates for the time delay by adopting a new variable d in addition to the 3D position coordinates x, y, and z [6]. Still, this method assumes an equal processing delay *d* for all receivers, which rarely happens in reality.

Also, to enable the indoor localization of mobile devices without any special equipment, the system uses an audible sound instead of an ultrasound (a 4.01 [kHz] tone with 0.2 [s] of duration is emitted as the pulse signal). This sound is not only prone to interferences from ambient noise, but can also be perceived by (and irritating to) most people.

3. Features of Sonicstrument

As mentioned above, the Sonicstrument addresses these issues with the following key features:

3.1 Doppler Effect

The Sonicstrument measures the Doppler shift of beacon signals. Compared to the aforementioned TOF approach, this method is more robust to ambient noise. Also, it is independent of the irregular temporal delay of the platform, while the TOF method is vulnerable to the delay. Furthermore, when the temporal resolution and audio bandwidth is limited, the Doppler shift analysis is expected to provide a better "resolution" for detection than pulse reflection analysis [1, 8]. In addition, since this method does not require any interval between the acoustic signals, the system is suitable for continuous detection, whereas the TOF method should wait for reverbs to decay.

3.2 Frequency Bandwidth

Similar to [6], the Sonicstrument also utilizes a non-ultrasonic, audible sound, but it also focuses on a different frequency range that is rarely perceived by most users. Typical acoustic transducers (e.g., everyday earphones and microphones) cover a frequency response range from around 20 [Hz] to above 22 [kHz] [5], thereby spanning the human audible range. Still, even within this range, there are frequency bands (both high (above 18 [kHz]) and low (below 60 [kHz])) that are practically inaudible for most people at usual loudness levels. These "marginal" areas can be utilized for motion detection with virtually no disturbance or overlap against the sonic "contents." At the same time, using these frequency ranges allows for easy implementation of the system, as described below.

4. SYSTEM DESIGN

4.1 Overview

This system uses a laptop (Dell Inspiron 1420) as its platform; it is equipped with an internal stereo microphone and outputs a maximum of 5.1 channel audio (a common feature with most personal computers these days), among which we use two

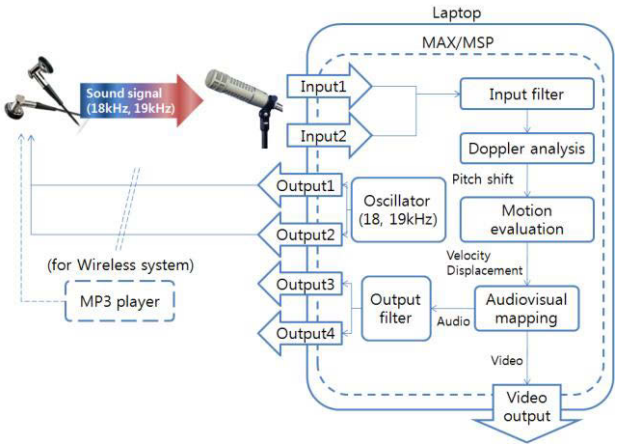


Figure 1. System block diagram of Sonicstrument system

channels for control signal output and other two for final sound output. The earphones that act as a controller are connected to the first two channels that transmit high frequency sound waves. Therefore, the system requires no extra device except for one computer. (For use on stages, the sound can be emitted from a commercial MP3 player, which can help in keeping the performer wire-free. This will be discussed later.) As the user moves his hands with the earphone, the Doppler shift occurs on the signal sound wave. The internal stereo microphone receives this distorted sound wave, while the MAX/MSP software analyzes the Doppler shift of the sound and calculates the hand motion in a normal direction for the microphone. Finally, the motion data are mapped to show visual and audio output.

This methodology is basically identical to the aforementioned ultrasonic sound tracking in mechanism (which is also used for human motion detection). However, this system uses stereotypical microphones that can handle the whole bandwidth of "theoretical" human hearing, which makes it necessary to use audio filters for noise elimination in most of the audible range. Furthermore, the final sound output from the system should not overlap the control signal bandwidth in order to have no interruptions.

4.2 Frequency of Control Signal

First of all, there are two choices of frequency. The frequency can be lower than the practical audible range or higher. A higher frequency range was chosen for two reasons. First, common room noise is distributed in a lower frequency range than a higher one. By using a higher frequency range, the system gets less influence from these noises. This higher noise-immunity is desirable for this system because it is more reliable when common earphones and microphones with lower precision are used. Second, a higher frequency yields a higher Doppler shift resolution, as the frequency changes more for the same hand velocity.

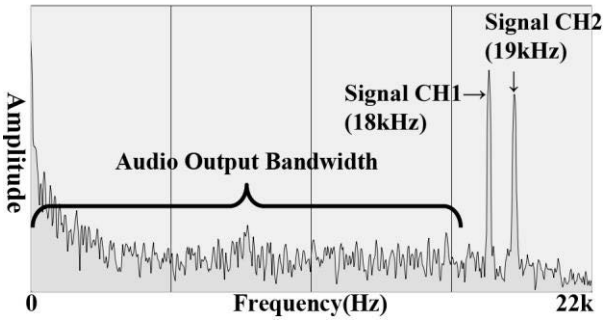


Figure 2. Spectrum distribution of control signal and audio output

Again, most of the commercial earphones cover a frequency range from around 20 [Hz] to around 22 [kHz]. Some kinds of earphones cover a narrower range around 20 [kHz]. This is the upper limit of frequency that can be used for this system. Also, the frequency range should not get too low to avoid being perceived by the user.
For these reasons, the sinusoids of 18 [kHz] and 19 [kHz] are used as the control signals for the left and right earphone respectively.

4.3 Signal Processing

4.3.1 Input Filter

The first step for the control signal that the microphones received is the band-pass filter. Because the Doppler shift is detected by a peak-tracking module, all of the noise peaks outside of the controller frequency need to be eliminated. To pass the two control signal ranges through while cutting off other ranges as much as possible, two narrow band-pass filters are connected in parallel.

4.3.2 Doppler Analysis Module

The signal that is able to pass through the filter is then sent to a Doppler analysis module where the 'fiddle~' object takes the biggest role. 'fiddle~' is a Max/MSP external object by Miller Puckette that tracks down multiple peaks and returns their frequency and amplitude in real-time. We already know the control signal is 18 [kHz] and 19 [kHz]. By comparing these reference frequencies to the detected peak frequencies, the motion velocity, which is a function of the frequency ratio, is evaluated. And the displacement is also numerically evaluated with this velocity data.

4.3.3 Output Filter

Finally, the evaluated motion data are connected to a proper visual or audio reaction. Before doing that, we have to consider that the audio output and control frequency are both in the audible range. To prevent the risk of any interruptions to the control signal, audio output should pass through a low-pass filter.

5. PERFORMANCE TEST

First, a qualitative test was carried out to see if the system can stably trace the motions of the handheld earphones (a demonstration video – titled as 'Video 1' – is available at [14]). The system successfully traced the motions of the earphone very smoothly and with no interference from ambient noises and between two control signals.

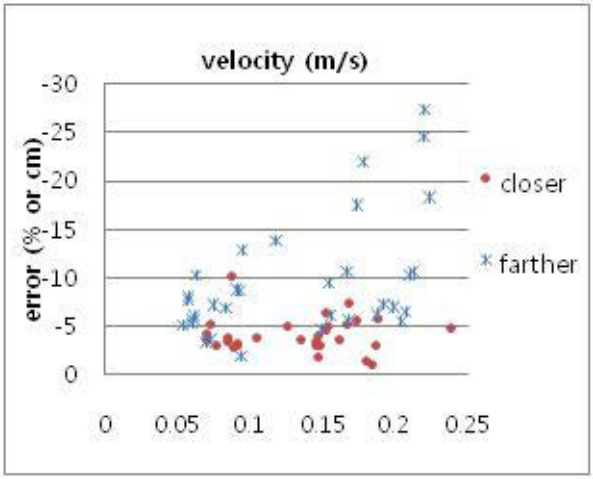


Figure 3. Error distribution of Sonicstrument from a performance test

In addition, a quantitative experiment was conducted to see whether the system can properly evaluate the displacement of the earphones. This was done by moving an earphone for 1 [m], and comparing the system-measured distance with real displacement. There were two criteria: direction (farther or closer) and the velocity of the motion. Datasets were collected thirty times for each direction, while the velocity values were randomly distributed.
Figure 3 shows the results; the mean and standard deviation of the measured values for each direction are -4.16 [cm] / 1.81 [cm] (closer) and -9.58 [cm] / 6.37 [cm] (farther). We can also see that the error distribution tends to increase when the direction is farther away and the velocity is faster. In our applications, the system could detect the motions of the user with reasonable precision and robustness. However, depending on the application, this error may not be neglegible and, since this aspect may be related to the frequency resolution, increasing the FFT window size may solve this problem and enhance the accuracy. Also, increasing the sampling rate would help reducing the latency due to the FFT and detecting more slight frequency shift in high frequency signal.

6. DEMONSTRATIONS

6.1 Musical Instrument

The first application of the system was on a computational musical instrument, which was exhibited at Anthracite, Seoul, Korea in 2010 (a demonstration video – Video 2 – is available at [14]). We took the violin as the metaphor for this motion-sound mapping; for a right-handed violinist, the left hand presses the strings to determine the pitch and the right hand bowing action excites the strings to generate the sound and control the volume. In our case, displacement of the earphone on the left hand from the microphone was mapped to the pitch (1 scale per 0.1 [m]) and the velocity of the earphone on the right hand corresponded to the audio gain.
As its platform, the system used the same laptop PC that we used for basic system implementation. We used two channels for control signal output, and two other channels for sound output.
The output sound was generated in real-time using subtractive synthesis. Multiple filters sculpted a pink noise to make a violin-like sound, while filter coefficients were manipulated to control pitch. Also, as mentioned in *4.3.3*, the output is filtered to prevent the interference with the control signal.

In this performance, the system successfully functioned as a musical instrument; pitch and gain values were controlled as the user intended. One problem, however, was the error accumulation of the estimated position of the left hand. To compensate for this, a reset function was implemented to be triggered when the gain from the right hand became large enough.

Figure 4. The first demonstration of Sonicstrument (musical interface & visualization)

Figure 5. The second demonstration of Sonicstrument (Interactive performance '4nm')

6.2 Interactive Performance

The second application of the system was at an interactive dance performance. As sonicstrument can work at a distance of about 10 [m] maximum, it can be utilized for most small-sized theater performances. In order to make the system wireless (which is critical for devices used in active dance performances), a portable music player was used to generate a control signal instead of PC; the music player was attached to the performer's body, and the earphone transmitters from the music player were held by the performer. Also, instead the internal microphone of the laptop in previous case, an external microphone was placed behind the curtain on the side of the stage to detect the control signals.

This system enabled us to detect the performer's hand motions or changes in body position. Measured control inputs were mapped to appropriate audiovisual stage effects; in a piece called *4nm*, the velocity data triggered a water flow sound and visual distortion effects.

Through this setup, sonicstrument showed its potential as an easy-to-use and highly effective interactive device for larger-scale performance. A video footage from this performance (Video 3) is also available at [14].

7. CONCLUSION

Sonicstrument is aimed at providing a virtual motion-detection interface without extra sensor devices. By using a sinusoid signal at an audible range, the commonly used earphones are utilized as a signal transmitter. The signal frequency is technically in the humans' audible range, but it is an extremely marginal area that most humans cannot perceive, thereby enabling continuous detection. As the system does not require any extra sensor device, individual users can easily own their tangible interface, and it can be simply applied to the performing arts.

This system uses a Doppler analysis instead of dominant TOF method. This is advantageous for reliability in a noisy environment. There is also a higher resolution that is not constrained directly by the system's time resolution and irregular time delay, while the TOF method has these restrictions. This is another good trait for popular interface.

In contrast, we were able to reconfirm the inherent limitation of the Doppler analysis in displacement estimation: error accumulation. Future work to compensate for this limitation can use the combination of the TOF and Doppler analysis method.

Also, for an environment with multiple microphones, like the laptop that we used that has 2 microphones, the direction-of-arrival (DOA) estimation technique for audible sound [2] can be adapted and it is expected to increase the degree of freedom in the system.

8. REFERENCES

[1] Amundson, I., Koutsoukos, X. and Sallai, J. Mobile Sensor Localization and Navigation using RF Doppler Shifts. In *Proc. the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, ACM Press (2008), 97-102.

[2] Chandran, S. 2006, Direction Estimation of Broadband Sources for Auditory Localization and Spatially Selective Listening, *Advances in Direction-of-Arrival Estimation*, Artech House, Boston, pp. 305-326.

[3] Dogaru, T., Le, C. and Kirose, G. Analysis of the Radar Doppler Signature of a Moving Human.

[4] Harter, A., Hopper, A., Steggles, P., Ward, A. and Webster, P. The Anatomy of a Context-Aware Application. *Wireless Networks 8, 2* (2002), 187-197.

[5] Headphone – Wikipedia http://en.wikipedia.org/wiki/Headphones

[6] Lopes, C.V., Haghighat, A., Mandal, A., Givargis, T. and Baldi, P. Localization of Off-the-Shelf Mobile Devices Using Audible Sound: Architectures, Protocols and Performance Assessment. In *Proc. SIGMOBILE 2006*, ACM Press (2006), 38-50.

[7] Motion Detection – Wikipedia. http://en.wikipedia.org/wiki/Motion_detection.

[8] Paradiso, J., Abler, C., Hsiao, K. and Reynolds, M. The Magic Carpet: Physical Sensing for Immersive Environments. *Ext. Abstracts CHI 1997*, ACM Press (1997), 277-278.

[9] Priyantha, N.B., Chakaborty, A. and Balakrishnan, H. The Cricket Location-Support System. In *Proc. MobiCom 2000*, ACM Press (2000), 32-43.

[10] Reynolds, M., Schoner, B., Richards, J., Dobson, K. and Gershenfeld, N. An Immersive, Multi-user, Musical Stage Environment. In *Proc. SIGGRAPH 2001*, ACM Press (2001), 553-560.

[11] Seniuk, A. and Blostein, D. Pen Acoustic Emissions for Text and Gesture Recognition. In *Proc. 10th International Conference on Document Analysis and Recognition*, IEEE (2009), 872-876.

[12] Vlasic, D., Adelsberger, R., Vannucci, G., Barnwell, J. and Markus G. Practical Motion Capture in Everyday Surroundings. In *Proc. SIGGRAPH 2007*, ACM Press (2007).

[13] Zhang, Z., Pouliquen, P.O., Waxman, A. and Andreou, A.G. Acoustic micro-Doppler radar for human gait imaging. In *Journal of the Acoustical Society of America Express Letters*, Vol. 121, No. 3(2007), pp. 110-113.

[14] Video clips of Sonicstrument demonstrations, http://aimlab.kaist.ac.kr/~badclown/Sonicstrument

Solar Sound Arts: Creating Instruments and Devices Powered by Photovoltaic Technologies

Scott Smallwood
University of Alberta
Fine Arts Building 3-82
Edmonton, AB T6J4H1
Canada
scott.smallwood@ualberta.ca

ABSTRACT

This paper describes recent developments in the creation of sound-making instruments and devices powered by photovoltaic (PV) technologies. With the rise of more efficient PV products in diverse packages, the possibilities for creating solar-powered musical instruments, sound installations, and loudspeakers are becoming increasingly realizable. This paper surveys past and recent developments in this area, including several projects by the author, and demonstrates how the use of PV technologies can influence the creative process in unique ways. In addition, this paper discusses how solar sound arts can enhance the aesthetic direction taken by recent work in soundscape studies and acoustic ecology. Finally, this paper will point towards future directions and possibilities as PV technologies continue to evolve and improve in terms of performance, and become more affordable.

Keywords

Solar Sound Arts, Circuit Bending, Hardware Hacking, Human-Computer Interface Design, Acoustic Ecology, Sound Art, Electroacoustics, Laptop Orchestra, PV Technology

1. INTRODUCTION

The phenomenon of photovoltaics (PV), the conversion of light energy into direct current, was initially discovered in 1839 by the French physicist Alexandre-Edmond Becquerel. Over the next hundred years, experiments in photovoltaics progressed slowly, and it wasn't until the 1950s that the technology began to be used in earnest for practical harvesting of energy in the space industry through sustained research at Bell Laboratories, AT&T, and Western Electric. The 1970s saw the mass production of solar panels, but the technology failed to progress rapidly and take hold as a viable alternative to fossil fuels due to powerful oil company lobbyists and a lack of government subsidy support. Within the last ten years, however, PV technologies have emerged as a real viable alternative, particularly in remote installations, as the many solar panels along roadsides and gas lines powering pumps and hazard lights attest.

The last ten years has also seen a rapid increase of the use of PV technologies in art, particularly art installations installed in remote locations, such as the Black Rock Desert in Nevada during the annual Burning Man Festival. As well, it seems that the various ecological threads of sustainability, including music's own acoustic ecology phenomenon, has created an environment in which outdoor, site-specific arts practice has grown into a real force of interest among many artists and curators. With an ever-emerging culture of DIY electronics and media art forms, it is not surprising that PV would offer attractive solutions to remote power needs, but it also offers a interesting new variable into the possibilities of environmental interaction. This paper will introduce some ideas towards putting PV to work in experimental sound works and instruments, with emphasis on environmental interaction through PV variability.

2. BACKGROUND

In 1979, Alvin Lucier, in collaboration with electronic designer John Fullemann, completed his solar-powered sound installation entitled *Solar Sounder I*. The piece was installed as a semi-permanent installation in the lobby of City Savings Bank in Middletown, CT. The solar panels were placed at ground level such that a person could cast a shadow over a panel and cause the sound to change. Speaking about this piece, Lucier tells us that the idea behind the piece is not that the audience can manipulate the sound, but that "the sound is changed by the rotation of the earth and the revolution of the sun, and so changes with the seasons" [5]. As the sun fell on the cluster of PV panels in different proportions based on the angle of the sun, which if course is dependent upon rotation and revolution, it caused the piece to sound in a distinctive way during that particular time of day and part of the year.

Craig Colorusso's *Sun Boxes* (2009-10) consists of twenty speakers with circuitry that play guitar loops of different lengths, each with independent solar PV systems, creating an evolving soundscape [1]. Like Lucier's work, this piece generates sound only when the sun falls on the solar panels, thereby linking the piece to real-time environmental factors. Jeff Federsen's *Earth Speaker* (2007), a series of PV-powered speakers and circuits, absorb sunlight during the day, and play low frequency sounds at night [6]. Nigel Helyer, a Sydney-based sculptor and sound artist, has created several interesting environmental works that are completely self-contained using PV technologies, including *Haiku* (2003), *Lotus* (2006), and *Meta-Diva* (2001), consisting of a group of loudspeakers on the ends of long aluminum pipes, sticking out of the water in a cluster. Each circuit is a solar-powered "voice," emitting short samples of the sounds of birds, frogs, insect sounds, etc, which are sounded at unique rhythms, creating a texture of natural voices [3].

These works all have different approaches to generating sound, but common among them is the way in which they relate to their environment, both in terms of their reliance on that environment's natural energy level, and in terms of the resulting soundscape. The approach to distributing power is partially responsible for this, having many self-contained, slightly individualistic objects that are part of a larger whole, but the sounding objects themselves utilize the variability of the power source in the sounds produced. In this way, the sounds

are linked explicitly to environmental energy factors. This approach to dealing with power issues has interesting implications not only for sound installation work, but also for sound performance, improvisation, and composition.

3. SOLAR OBJECTS AND INSTRUMENTS

The author's own inquiry into this research comes from a long-standing practice of soundscape composition and site-specific performance, as well as through technological research interests and experiences in interactive instrument design and laptop performance [2][8]. Some general principles of the following designs include:

- Self-contained systems. The objects and instruments are conceived as all-in-one devices, featuring their own sound generation circuitry, loudspeakers and energy collectors (PV arrays).
- Portability. The objects are highly portable, and in the case of the instruments, they are small hand-held devices.
- Context sensitive. The objects respond directly and immediately to light both in terms of power and sound control.
- Recycled materials. The objects are constructed from inexpensive parts and recycled materials.

What follows are descriptions of several devices and instruments that have resulted from this preliminary research.

3.1 Solar Noise Discs

These small, handheld instruments grew out of some earlier experiments [2], and all have the same basic features. They are packaged in recycled film canisters, particularly small ones intended to house 3.5-inch reels of 16mm film. They have speaker holes on one side, from which the sound emits. On the other side, they feature three photocells, and five body-contact points via Canadian dimes, which when touched, use body capacitance to affect changes in the circuit (see Figure 1). The sound-producing circuits in these instruments are based on Schmitt trigger logic chips such as the 74C14 and the 4093. These bistable multivibrators can be used to produce a square-wave oscillator by connecting an RC circuit between input and output pins on the chip. Several oscillators can be produced with a single chip, and circuit-bending techniques are used to create modulations and instabilities in the circuit. The resulting sounds are amplified through an LM386 amplifier and a small speaker built in to the instrument. The instrument includes a 2.1 mm coaxial power jack for DC power, which connects to a PV array, or to any DC power source for that matter.

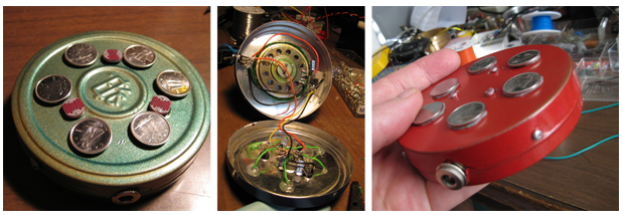


Figure 1: "Fly" Handheld Solar Electric Instruments

The instrument can be played by holding it between the hands like a tiny accordion. One hand becomes a kind of mute, which is essentially volume: to stop the sound, you just cover the holes, opening your hands up as you need more sound, which also changes the timbre. The other hand, of course, covers the three photocells and connects the body contacts in various ways, depending on how fingers are arranged. How one does this, and how it affects the sound is something that has to be experienced and practiced. It seems chaotic at first, but one soon learns that things can be controlled, and repeated, if conditions are just right. There are currently three of these instruments in existence, and a few oddball versions as well. Although they each sound different, due to variations in the circuitry, there is a quality about all of them that is similar due to the basic circuit design and the physical resonance properties of the case.

To power the instruments, there are two different "levels" of power, to allow for a couple of different types of lighting level requirements and volume levels. On sunny days, these instruments can be played with "solar flaps", which are essentially small solar arrays mounted on either side of a cardboard flap, which plugs into the instrument and kind of dangles, flapping in the wind if it happens to be windy (see Figure 2). This will supply a maximum of about 600 mW of power, more than enough to power the instrument quite sufficiently if its sunny, and it allows for the possibility of rapid changes of character since one can quickly move it in and out of shadows. The next power level is achieved by connecting the instrument to a larger solar panel that can be worn on a pack-back, or simply placed on the ground. The Go Power! DURAlite series of PV panels, manufactured by the Carmanah Technologies,[1] have proven to be the most effective solution in the field. These lightweight panels are designed for RV and marine usage, and they are quite robust due to the fact that the solar film is encased in a fiberglass laminate, making them virtually unbreakable, as well as waterproof and resistant to clouding by overexposure to intense sunlight. These products come in three varieties: 5, 10, and 20 watt panels, each with an operating voltage of 16 volts. Even in cloudy conditions, the 5 watt panel has proven to work very well.

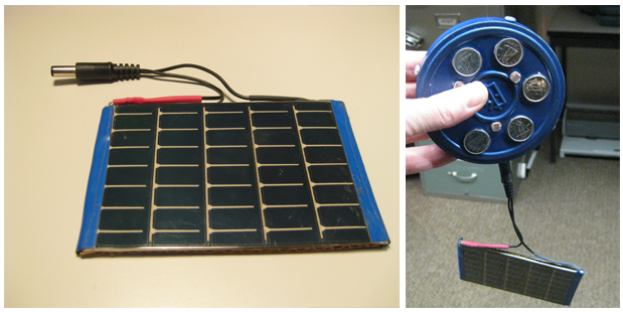


Figure 2: "Solar Flap"

3.2 Bird

This installation project was an attempt to create something that was completely self-contained, with the PV technologies being "built-in" to the object. Using small 500 mV cells generating about 17 mA of current, a ring of several of these were connected in series to form an a PV array, which were glued onto a small disc. This array, of course, failed to generate enough wattage to power the amp-speaker circuits used in the instruments above, but would work just fine powering piezo-elements. Sound generation was created using similar Schmitt trigger circuits to the instruments above, connected to piezo-speakers. The results are a series of beeping, twittering, and buzzing devices that are completely self-contained, each using about 40 mW of power.

In preparing for this work, the work of Ralf Schrieber and his "solarsoundmodule" designs were quite inspirational [7]. The key interest in this work was not so much in the individual design itself, but in the idea of several of these operating as a

[1] http://www.carmanah.com

series of voices in a single system. It became interesting to imagine walls covered with these devices, in clusters or spread throughout a space, or perhaps hanging from the ceiling in mobiles.

Figure 3: *Bird* (2009)

The resulting piece is indeed a mobile, named *Bird*, which consists of four round objects based on the 4093 chip, a piezo-speaker, a bottle-cap resonator, and a ring of seven small 500 mV solar cells in series, all housed in plastic juice bottle caps (see Figure 3). The four devices hang in a column, connected one to another by ball bearing fishing swivels, allowing each to turn freely. These hang inside a column of streamers made from ¼" red plastic leader tape. This allows for the wind to affect the sounds from the devices due to rippling shadows cast by the streamers, as well as helping to generate the force necessary to cause the devices to turn in response to the wind, creating an ever-changing sound based on the sun and the wind. Bird has been exhibited on its own, and as part of larger installations.

3.3 Domeintonators

These devices are hybrid installation objects and instruments, and were designed using circuitry similar to that of "Bird" above: a 4093 NAND gate chip with two distinct patterns of modulating square wave oscillators, each of which is sounded by one of two piezo speakers. The circuitry is installed in recycled security camera domes, and includes an audio jack for amplifying the signal (in stereo) of the domes if desired (see Figure 4). The devices also include two photocells, which are installed in the dark battery compartment in the bottom of the housing. When the door to the battery compartment is removed, the performer can manipulate the sound of the instrument by "playing" the photocells underneath the instrument, thus causing changes in the speed of the two modulated square waves.

These are powered by a 150 mW flexible solar strip by PowerFilm, Inc., which manufactures a variety of small to medium-sized flexible strips.[2] The solar strip is mounted in an arch over the circuitry inside of the clear dome top of the instruments. This makes the instrument completely self-contained, but it is expandable in the sense that it can be amplified using external equipment if desired.

Figure 4: "Domeintonators"

[2] http://www.powerfilmsolar.com

These instruments have been used in performances as well as in installation settings, but ultimately these might be best suited to an interactive installation in which the existing six devices could be "played" by listeners, while being further amplified by an additional solar-powered amplifier and loudspeaker, making the six instruments function as one larger, organ-like instrument.

3.4 Arcade Bells

The Arcade Bells resulted from a desire to move beyond these analog-ish methods for making sound by employing a small microprocessor. In addition, this piece uses piezo discs to explore ways to "ring out" or resonate larger objects, such as metal sheets, bowls, and other devices that might make nice resonators. This particular piece uses four small copper-plated footless goblets as transducers by gluing large piezo discs to the bottom of each and sending the leads through a hole drilled through the center. A housing container is attached to the bottom, made from recycled evidence containers, in which the circuitry resides.

Figure 5: *Arcade Bells* (2010)

The circuitry consists of an ATMEL ATmega328 chip and its life support (voltage regulator and power capacitors, crystal clock, etc), a LM386 amplifier, a small transformer, and a photocell, which sticks out of the cup like a small flower. The chip is programmed to generate square waves at audio rates in patterns; specifically, ascending spectral arpeggios, similar to 1980s arcade game sound effects. The chip also monitors the state of the photocell, which when bright, means the spectral arpeggios get faster, and open up to more of a range. The sound created by the chip is amplified through the LM386, and impedance-matched to the piezo-disc, which then resonates through the brass cup.

Using the ATMEL chip requires more power than the simple logic chips used in previous experiments, but nevertheless they are quite capable of operating in low power states. This entire circuit consumes less than 500 mW and can operate with much less. The amplifier is powered outside of the 5 volt regulator's loop, allowing it to be more directly effected in volume by the full voltage coming in from the solar panel.

3.5 Solbutter Instruments

The final series of devices discussed here are a new series of microprocessor-based devices, based in part on the work done on the Arcade Bells above. These instruments also use the ATMEL ATmega328 chip to generate sound, as well as to read sensor data from various control sources. As of this date, two prototypes have been completed, both of which use the ATMEL's timer interrupts to generate waveforms other than square wave forms, including sine waves and FOF-based granular synthesis algorithms.

Like the noise disc instruments above, these are round, handheld instruments with a built-in speaker amplified by a LM386 chip, mounted on the bottom of the container. The instrument shown (figure 6) features two pushbuttons (taken from a recycled toy), a 50K ohm potentiometer, and four photocells.

Figure 6: "Solbutter I"

Code-wise, the sound producing methodology, written in the Arduino environment, uses the chip's PWM (pulse-width modulation) digital output capabilities to simulate analog waveforms, and also uses a timer-interrupt routine to ensure that the sound-producing part of the code gets priority. This device uses code based in part on the open-source "Auduino" project by Peter Knight, which implements a simplified FOF synthesis model to create a low-fidelity granular synthesis architecture [4].

Powering this instrument is accomplished the same way that the noise disc instruments are powered: either by plugging in a "solar flap", or a larger solar panel (or a battery or any other DC power source). This instrument requires at least 250 mW, and includes a voltage regulator and filter for the chip. The amplifier uses the raw voltage of the solar panel, meaning that the relative volume level is proportional to the power received from the PV source.

Figure 7: Performing with Tealfly in the Ravine

4. FUTURE WORK

While these humble beginnings have been instructive, it seems clear that this research has tremendous potential. It has already led to a number of interesting performances and installations, and it is my hope that through a combination of these obsessions and experiments, and the obsessions and experiments of others, a kind of "solar sonics practice" might emerge, which might open up a whole new world of site-specific and spatial performance concepts. In the words of Alvin Lucier: "Composers are thinking now of a timeless kind of depth; that is, of creating and going *into* a sound-space, rather than moving horizontally *along* it" [5]. This, to me, is what much of this work is about, and where much of its potential lies.

In addition, this work has tremendous potential for artists involved in the field and philosophy of acoustic ecology. In addition to the obvious utility of creating electronic music performances outdoors using sustainable technologies, the metaphor of real-time energy production as a sonic parameter is a powerful one, and will undoubtedly result in a variety of interesting sound work with explicit connections to environmental factors.

The future will bring a new series of installations, some of which will be prototyped and evaluated at the Burning Man Festival in 2011. Also planned is a PV-powered portable hemispherical speaker for supporting mobile outdoor performances with laptops and other electronics that require additional amplification. Finally, I intend to infect more people with this bug through instrument building workshops and skill sharing, both here in my community and beyond.

5. ACKNOWLEDGEMENTS

I would like to thank Princeton University and the University of Alberta for their support of this research, as well as the MacArthur Foundation, GRAND[3], and the Canada Council of the Arts. In addition, the following people have been instrumental in their assistance and support: Perry Cook, Dan Trueman, Geoffrey Rockwell, Nicolas Collins, Mark Young, Gary Joynes, and Todd Janes.

Finally, I would like to acknowledge some of the manufacturers of the PV products that have been used in these experiments, particularly Carmana Technology Corp. and PowerFilm, Inc. In some ways, PV is still just getting off the ground, and the number of manufacturers of these products are small, mostly centered on products for large-scale power installations or home conversion systems. PowerFilm in particular is developing a wide variety of PV solutions that are especially of interest to sound artists and designers, since they are scalable, extremely robust, and easy to mount on any surface with foam tape or adhesives.

For more information on this project, including more details, photos, schematics, sound samples, and video clips of these and other instruments and devices, please visit the project website at:

http://www.ualberta.ca/~ssmallwo/see/.

6. REFERENCES

[1] Colorusso, C. *Sun Boxes*. http://www.designboom.com/weblog/cat/8/view/10493/craig-colorusso-sun-boxes.html, 2010 (accessed Jan. 02, 2011).

[2] Cook, P, and Smallwood, S. SOLA: Solar orchestras of laptops and analog. *Leonardo Music Journal* 20 (2010).

[3] Helver, N. *Sonicobjects*. http://www.sonicobjects.com/ (accessed Jan 02, 2011).

[4] Knight, Peter. *Auduino*. http://code.google.com/p/tinkerit/wiki/Auduino, 2008 (accessed Jan 02, 2011).

[5] Lucier, A., and Margolin, J. Conversation with Alvin Lucier. *Perspectives of New Music* 20, 1/2 (Autumn 1981), 50-58.

[6] Mahajan, N. Art meets Solar Energy. http://www.ecofriend.org/entry/art-meets-solar-energy/, 2007 (accessed Jan. 02, 2011).

[7] Schreiber, R. Solarsoundmodule. http://www.ralfschreiber.com/solarsound/solarsound.html, 1996 (accessed Jan 02, 2011).

[8] Smallwood, S., Cook, P., Trueman, D., and McIntyre, L. Don't forget the laptop: a history of hemispherical speakers at Princeton, plus a DIY guide. In *Proceedings of the 2009 New Instruments for Musical Expression Conference*, Pittsburgh (2009).

[3] http://grand-nce.ca/aboutgrand/profile.html

An Approach to Collaborative Music Composition

Niklas Klügel, Marc René Frieß, Georg Groh
Technische Universität München
Boltzmannstraße 3
85748 Garching, Germany
{kluegel|friess|grohg}@in.tum.de

Florian Echtler
Munich University of Applied Sciences
Lothstraße 64
80335 München, Germany
florian.echtler@hm.edu

ABSTRACT

This paper provides a discussion of how the electronic, solely IT based composition and performance of electronic music can be supported in realtime with a collaborative application on a tabletop interface, mediating between single-user style music composition tools and co-located collaborative music improvisation. After having elaborated on the theoretical backgrounds of prerequisites of co-located collaborative tabletop applications as well as the common paradigms in music composition/notation, we will review related work on novel IT approaches to music composition and improvisation. Subsequently, we will present our prototypical implementation and the results.

Keywords

Tabletop Interface, Collaborative Music Composition, Creativity Support

1. INTRODUCTION

The adoption of electronic devices into music composition, performance and perception since the middle of the last century culminated in electronic music being part of today's popular culture. At the same time, new forms of music composition were introduced to a larger group of people, thereby gradually popularizing IT supported music composition. With this paper we want demonstrate how music composition can be transformed to become a more situative and collaborative process. We emphasize the discursive exploration of expression and exchange of musical ideas, thereby focusing on experts in the field of electronic music composition (cp. [2]). In this context it is worthwhile stating that we restrict ourselves to "electronic composition of electronic music", where "electronic composition" means that musical elements are solely manipulated with computers and "electronic music" that all sounds are synthesized by a computer as well.

Unfortunately, most support tools for composing electronic music collaboratively focus on distributed and asynchronous collaboration. The attention on distribution and single-user user-interfaces poses a problem in view of designing truly collaborative applications as "*the user's concentration has to shift away from the group and towards the computer*" [9]. This contrasts to the importance of the social context and social interactions for creative tasks [4]. Face-to-face interaction in co-located settings has advantages over electronically mediated interaction such as the visibility of action and the ability to use verbal and non-verbal communication. As novel means of IT-support for performing music face-to-face, tabletop based applications such as the reacTable [13] are promising, as they particularly support this kind of social context. While in the single user case of creatively manipulating music on a stand-alone PC, every action can be arbitrarily stored, undone, resumed and reflected on, the improvisational social case lacks these possibilities for obvious reasons. Considering these aspects, the research question to be answered within this paper is:

> *How can the advantages of co-located collaborative musical improvisation and single-user oriented applications for systematic and iterative music composition be combined in tabletop-based system that synchronously, collaboratively and in a co-located way allows for the composition and performance of music?*

In order to devise an application suited to answer this question, at first requirements for applications to support (group) collaboration on such devices in general have to be discussed. Afterwards we will discuss some paradigms of music composition in general and will review related work in regard to distributed as well as co-located IT systems for music composition and performance. Finally the application will be described in detail.

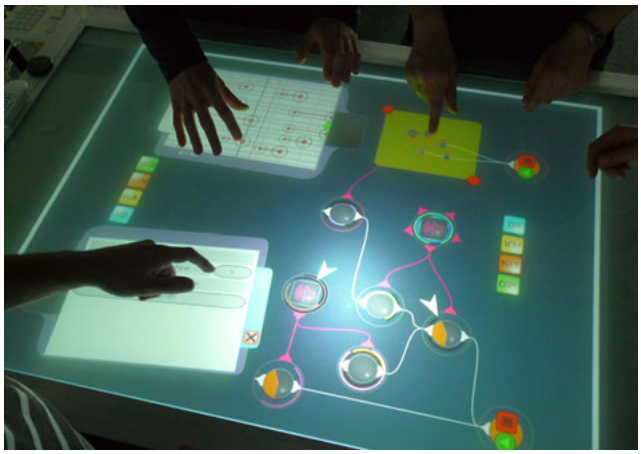


Figure 1: The Application in Use

2. GROUP COLLABORATION

Multi-touch tabletop devices feature several characteristics that support collaboration as they feature intuitive methods of shared and parallel input. Gestural (and tangible) interfaces exploit the human kinesthetic capabilities gained from life-long interaction with the physical environment [10]. This results in both a rich vocabulary for Human Computer Interaction and the reduction of cognitive overhead for tactile interaction [5, 11]. These fundamental properties for tabletop devices overlap in their consequence with properties desired and essential to collaboration: Barriers in

NIME'11, 30 May–1 June 2011, Oslo, Norway.

the computer mediated interaction between collaborators are alleviated and the immediate joint use of interaction is fostered. While these characteristics provide a basis for enabling situated collaboration, a practical realization is bound to the domain of application and design principles solving high-level issues. In this regard, the following questions arise: How to support the complex dynamics of collaboration? Which constructs in the application support a high degree of synchronicity in the collaboration?

While it is all but impossible to grasp collaboration holistically, there are several aspects in collaboration that have been identified to play a key-role for realizing applications. In the field of Computer Supported Collaborative Work (CSCW) those are: **group awareness**, **group articulation** and **tailorability** [3, 6]. **Group awareness** describes the ability of individuals in the group to share and gain mutual knowledge about the activities performed. Hence, a good sense of group awareness helps collaborators to coordinate activities and to access shared resources by simplifying communication. **Group articulation** refers to the ability to partition the group task into units that can be combined, substituted and reintegrated in the group task, while **tailorability** proclaims the ability for individual adoption of technology (in unintended ways).

Concerning tabletop interfaces, various forms of **coupling** can be identified [18]. Coupling describes the different styles of the mediated collaboration around the tabletop device and as such it results in a steady flux of the group configuration. Supporting such group transitions is the key to permitting the dynamics of collaboration to happen, since they reflect a natural style of communication and interaction between collaborators. The spatial usage of the tabletop environment for personal and group tasks (**territoriality**) is also relevant. It helps to coordinate actions with artifacts or with collaborators on the tabletop [17] akin to the human division of space in the physical reality. In general, the parallel interaction with the application are very important. On one hand, they foster group awareness and group articulation since centralized controls may disrupt the workflow [14, 10]. On the other hand they make contributions for all group members more democratic because of the distributed right to control artifacts. In this way, individual users are not prioritized and thresholds for collaborating are lowered [7]. These aspects will serve as the basic requirements for our application design. How we explicitly address these issues like territoriality and coupling will be discussed in section 5. Before, the application field of music composition as well as related work needs to be regarded.

3. COMMON PARADIGMS IN MUSIC COMPOSITION & NOTATION

It is important to briefly discuss some basic concepts of music composition, notation and the relevant prerequisites for collaboration in the domain of this application to further frame the requirements for the design and its musical applicability.

A key aspect for composition is to establish a notational system to express the comprising musical events. Since the sonification of the composition is performed by the computer, its interpretation and execution by a performer is omitted. Thus, the control over the perceptual dimensions such as timbre is part of the notation and thereby the composition. Therefore, for electronic music the notation serves as a blueprint for the stream of sound - tailored to the specific requirements of musical expression for a particular piece. But especially in the context of collaborative use of the notation, the problem is to create a musically expressive set of notational symbols whose meaning and effect is plain to all participants. Furthermore the notation must support collaboration in a concurrent and simultaneous way.

Traditional music admits various organizing principles, both hierarchical and non-hierarchical, making the communication of musical ideas a knowledge representation problem. In a collaborative setting, it is essential for musical expression to support such organizing principles and to make the compositional structure modifiable and, for the dialogue, holistically comprehensible. But this is conceptually difficult to realize for a shared tabletop interface. It is caused by the representation (see section 5.3 - User Interface) and the rigid structure of a (traditional) score for the collaborative context. In many cases, the development of musical ideas into the resulting final composition is inherently bound to properties that have been chosen at the beginning. Insofar the process of composing itself is constrained by assumptions taken gradually but on a global level (e.g. the instrumentation of a piece, tempo changes) while the locality, the independence of (sonic and temporal) characteristics for outlining a musical idea is not supported. An application for collaborative music composition therefore has to allow the creation of meta-/intermediate arrangements of a composition - a conceptual unification of creating arbitrary musical sketches (separate arrangements) that can coexist and be gradually shaped into a final composition/arrangement. In view of enriching the overall workflow, solving this problem of conceptional unification is the key point in a collaborative environment. In spite of the high degree of complexity inherent in the process of composing, several application supporting electronic performance as well as composition of music have already emerged. Some tabletop applications will be presented next.

4. RELATED WORK

Tabletop interfaces reflect one end of the spectrum of synchronicity for supported group interaction. Interaction is carried out physically and locally using a single shared workspace. In this regard, many tabletop music applications share goals with ours. Their field of application widely ranges from multi-user instruments (e.g. reactTable [13]) to probabilistic composing environments (e.g. Xenakis [1]). Some of them use physical objects (**tangibles**) that represent artifacts or functionality of the application. As they act as a bridge to the digital representation of an object they make use of our kinesthetic and spatial intelligence for interacting with the application. Thus, tangibles are by concept excelling implementations of the direct manipulation metaphor. Group awareness is assisted as all information that is conceived for interaction with the application is physically apparent, theoretically with all hidden states removed.

ReactTable [13] is a prominent example of such applications. It is a multi-user instrument portraying a modular synthesizer where the signal processing can be altered by gestural and physical interaction using tangibles. The signal flow is represented by the topological relationship between artifacts. As an extension to the original reactTable, scoreTable [12] allows the composition of looped phrases using tangibles while following a spatial approach to composition: the positions where objects are placed determine their values and their functionalities.

Conceptually, tangibles pose problems that are disparate with our requirements. In essence, the hard object relationship (virtual to physical) defeats arbitrary application of tangibles. First, multi-modality is difficult to express with physical objects. In a broader sense, it is challenging to superimpose concepts like context sensitivity on physical objects for an application. But they are essential if one wants to portray and rely on multiple object relationships. Second, it is difficult to represent hierarchical structures with physical objects in a one-to-one mapping. Furthermore, the complexity of a composition would be limited to the number and spatial properties of physical objects. Third, and most importantly, the common use of spatial relationships between tangibles to control the application is problematic for the group articulation as the group task cannot be arbitrarily partitioned into units and shared.

5. THE APPLICATION

We further frame the requirements for our application in regard to the user interface and the underlying data model:

5.1 Requirements

On the technical side, to support the dynamics of the collaborative interaction, the data model and operations have to fulfil these properties in terms of modifiability:

- Concurrency
- Radical, non-linear changes
- Real-time capabilities

For these properties to be perceived and utilized by users, the user interface has to commit to:

- Provide instantaneous feedback on the changes
- Provide consistent feedback on the state of the application / compositional structure and parameters of the synthesis
- Apply the aforementioned design paradigms and rough guidelines for supporting the group collaboration

To support the various aspects of Coupling and Territoriality the operational controls have to be independent of positioning. Furthermore, they should provide concurrent use without evoking ambiguous states and help reducing mode errors [16]. The latter is important, since the less the user's attention has to be drawn to controlling the application appropriately, the more the group communication and awareness is fostered.
We decided to split the user interface into two levels: the first level is for creating, modifying and controlling the compositional structure and the second one is for changing the properties of the elements in it.

5.2 Structures for Composition & Synthesis

The first level represents the compositional structure as a directed acyclic graph. It can be modified in real-time by the user interface. The graph is composed of two forests for the respective functional domains: the temporal order of musical events (sequence graph) and the audio synthesis (synthesis graph). The nodes in the first forest are sequences (cp. figure 2, 4a) which are usually short musical phrases, with the exception of the root nodes. Sequences can contain either arbitrary or chromatic control data (e.g. notes) for the synthesis. Here, the edges denote the succession of sequences; this constitutes the temporal order of an (meta-) arrangement. Each such arrangement is therefore represented as a single tree. The second forest is used to map the musical events of sequences to parameters of synthesizers. Therefore nodes in this representation are either sequences or synthesizers (cp. figure 2, 4b). Edges describe the flow of control information such as a sequence controlling pitch or frequency of a filter of a synthesizer. Parallel edges are allowed if a sequence is to control more than a single parameter of a synthesizer. The two forests are merged visually (sequence nodes exist in both forests so only one is visualized), as a result, the linear notion and depiction of instrument staves and a single timeline is circumvented. On the whole, this concept can be seen as a subset of a data-flow language that allows for graphical patching such as Pure-Data [15], although the application follows a different architecture internally.

Sequencing & Control of playback: A sequence is the atomic entity keeping track of musical events. It allows to insert and remove musical events and maintains their local temporal order. That is, all information regarding the timing of musical events is treated as relative to the beginning of the sequence (start and length can be changed). The type of succession expressed by linking nodes with edges can form either a sequential chain or branches of parallel sequences - this structure corresponds to a tree. The construction of parallel sequence chains equals several instrument staves with phrases - similar to traditional notation. The root node of a tree is a special object to control the playback of a sequence tree independently from others (cp. figure 2, 4c). It not only controls the immediate start and stop of the playback but also enables its synchronization to the global tempo and the looped playback.

5.3 User Interface

The visualizations of sequence and synthesizer nodes have additional graphical indicators : for sequences the current local playback position and for synthesizer the current signal volume generated. Naturally, edges are indicated by a line segment. They can be created by performing a dragging motion from one node to another one. The shared user interface provides areas on the borders, each for the specific type of nodes. Dragging from one of them to the desired position creates motion from one of them (cp. figure 2, 3) to the desired position. With the facility for users to arrange the visualization of the graphs freely and the techniques described thus far, the following conceptual gains for collaboration over traditional DAWs have been established:

- Building blocks of the composition can be freely moved around, grouped and interacted with regardless of the virtual position of artifacts
- Multiple arrangements/sketches of musical ideas are supported on the same interface. In fact there is no notion of a primary arrangement.
- Arrangements can be freely combined and rearranged.
- Changes on the compositional structure are immediately reflected visually and sonically

Manipulating Properties: The second level for interaction is used to change properties of nodes such as the type of the synthesizer or the musical events in a sequence. Chromatic musical events are manipulated using the piano-roll metaphor (common in modern DAWs). Setting properties of nodes in the graph structure is realized with the help of two elements in the user interface: the first one is for selecting the node (cp. figure 2, 1) whose properties are to be changed and the second one for visualizing and manipulating them (cp. figure 2, 2). The EditorView is similar to a window in the WIMP paradigm which floats above the compositional structure, but it can be rotated and translated arbitrarily to match the user's orientation. It also provides facilities to pan and zoom the visualization and to close the view. The Selector can be dragged onto a node in the graph structure. Synchronous to this, the contents in the EditorView are updated showing an interface to manipulate the selected node's properties. An arbitrary number of these Selector/EditorView pairs can be instantiated to edit the properties of multiple graph nodes concurrently. An integral feature is that the content in an EditorView again can be shared in a synchronous way, making the properties of a single item accessible for editing from several EditorViews. This allows users to concurrently interact and gain mutual insight in their doing, thus fostering group awareness.

Gestural Control: Most multi-touch capable tabletop surfaces do not associate touch points to a to a specific user. Here, collaborative gestural input can be prone to create ambiguous states in conjunction with multi-touch gestures. As a result, it was decided to utilize primarily single-touch gestures for interaction with objects that are intended to be shared. Multi-touch gestures were only applied for controlling the EditorView (to change the view and orientation). The limited amount of gestures is sufficient for the provided functionality while helping habituation (reducing cognitive overhead / fostering group awareness).

6. CONCLUSION

The control over the application is independent of the orientation or position of artifacts in the user interface. Ergo, the flexible and dynamic coupling of users while performing the group task is supported. This is achieved by avoiding spatial relations to express temporal and tonal dependencies of the composition with help of the graph visualization and is further extended with spatially independent and context-sensitive editors. Since all aspects about the modification and, not to be disregarded, the playback of the shared compositional structures can be controlled in real-time, the

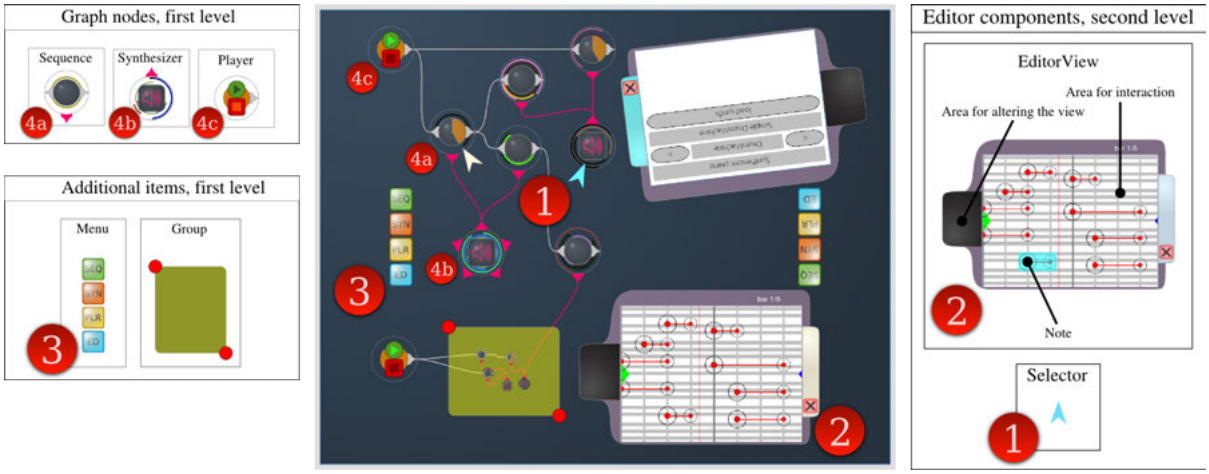


Figure 2: Overview of the Application's User Interface and Control Objects

discourse of musical ideas can be shifted from a verbally dominant one to a hands-on approach. Building blocks of compositional structures can be combined with instant sonic feedback. This simplifies the creative exploratory discourse while it blurs the line between composing collaboratively and improvising collectively:

- Users are encouraged to sketch and shape their musical ideas in real-time, synchronized with those that have been created or are in process of creation by others.
- The playback of disjoint arrangements can be triggered ad-lib by the users to fabricate new and derivative arrangements promptly.

In this paper an analysis of interface principles for collaborative co-located applications has been conducted. It has been shown that the dynamics and complexity of collaborative work imply numerous constraints on the design of our application. After the discussion of related work, the accompanied findings have been used as guidelines for the overall design of the prototype. In this way the implementation is rooted in user and collaboration centric design as opposed to building an interface on top of a preexisting application or framework to convey the inherent functionality. A demonstration video of the application can be seen in [8].

7. REFERENCES

[1] M. Bischof, B. Conradi, P. Lachenmaier, K. Linde, M. Meier, P. Pötzl, and E. André. Xenakis: combining tangible interaction with probability-based musical composition. In *TEI '08: Proc. of the 2nd Int. Conf. on Tangible and embedded interaction*, pages 121–124, New York, NY, USA, 2008. ACM.

[2] T. Blaine and S. S. Fels. Contexts of collaborative musical experiences. In *3rd Int. Conf. on New Interfaces for Musical Expression (NIME03)*, pages 129–134, May 2003.

[3] A. Cockburn and S. Jones. Four principles of groupware design. *Interacting with Computers*, 7(2):195–210, 1995.

[4] Mihaly Csikszentmihalyi. *Creativity : Flow and the Psychology of Discovery and Invention*. Perennial, June 1997.

[5] T. Djajadiningrat, B. Matthews, and M. Stienstra. Easy doesn't do it: skill and expression in tangible aesthetics. *Personal Ubiquitous Comput.*, 11(8):657–676, 2007.

[6] P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *CSCW '92: Proc. of the 1992 ACM conf. on CSCW*, pages 107–114, New York, NY, USA, 1992. ACM.

[7] H. Eden, E. Scharff, and E. Hornecker. Multilevel design and role play: experiences in assessing support for neighborhood participation in design. In *DIS '02: Proc. of the 4th conf. on Designing interactive systems*, pages 387–392, New York, NY, USA, 2002. ACM.

[8] M. R. Frieß, N. Klügel, and G. Groh. lzrdm: collaborative multi-touch sequencer. In *ACM Int. Conf. on Interactive Tabletops and Surfaces*, ITS '10, pages 303–303, New York, NY, USA, 2010. ACM.

[9] O. Hilliges, L. Terrenghi, S. Boring, D. Kim, H. Richter, and A. Butz. Designing for collaborative creative problem solving. In *C&C '07: Proc. of the 6th ACM SIGCHI conf. on Creativity & cognition*, pages 137–146, New York, NY, USA, 2007. ACM.

[10] E. Hornecker. A design theme for tangible interaction: embodied facilitation. In *ECSCW'05: Proc. of the 9th European Conf. on CSCW*, pages 23–43, New York, NY, USA, 2005. Springer-Verlag New York, Inc.

[11] T. Ingold. *Beyond Art and Technology: The Anthropology of Skill*. University of New Mexico Press, April 2001.

[12] S. Jordà and Marcos A. Mary had a little scoretable* or the reactable* goes melodic. In *6th Int. Conf. on New interfaces for musical expression (NIME06)*, pages 208–211, Paris, France, France, 2006. IRCAM — Centre Pompidou.

[13] S. Jordà, M. Kaltenbrunner, G. Geiger, and M. Alonso. The reactable: a tangible tabletop musical instrument and collaborative workbench. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 91, New York, NY, USA, 2006. ACM.

[14] M. R. Morris, A. Paepcke, T. Winograd, and J. Stamberger. Teamtag: exploring centralized versus replicated controls for co-located tabletop groupware. In *CHI '06: Proc. of the SIGCHI conf. on Human Factors in computing systems*, pages 1273–1282, New York, NY, USA, 2006. ACM.

[15] M. Puckette. Pure data: another integrated computer music environment. In *Proc. of Int. Computer Music Conference*, pages 37–41, 1996.

[16] J. Raskin. *The humane interface: new directions for designing interactive systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000.

[17] S. D. Scott, M. Sheelagh, T. Carpendale, and Kori M. Inkpen. Territoriality in collaborative tabletop workspaces. In *CSCW '04: Proc. of the 2004 ACM conf. on CSCW*, pages 294–303, New York, NY, USA, 2004. ACM.

[18] A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale. Collaborative coupling over tabletop displays. In *CHI '06: Proc. of the SIGCHI conf. on Human Factors in Comp. Sys.*, pages 1181–1190, New York, NY, USA, 2006. ACM.

A Reference Architecture and Score Representation for Popular Music Human-Computer Music Performance Systems

Nicolas E. Gold
University College London
Department of Computer Science
Gower St, London, WC1E 6BT, UK
n.gold@ucl.ac.uk

Roger B. Dannenberg
Carnegie Mellon University
School of Computer Science and School of Art
5000, Forbes Ave, Pittsburgh, PA, 15213
roger.dannenberg@cs.cmu.edu

ABSTRACT
Popular music (characterized by improvised instrumental parts, beat and measure-level organization, and steady tempo) poses challenges for human-computer music performance (HCMP). Pieces of music are typically rearrangeable on-the-fly and involve a high degree of variation from ensemble to ensemble, and even between rehearsal and performance. Computer systems aiming to participate in such ensembles must therefore cope with a dynamic high-level structure in addition to the more traditional problems of beat-tracking, score-following, and machine improvisation. There are many approaches to integrating the components required to implement dynamic human-computer music performance systems. This paper presents a reference architecture designed to allow the typical sub-components (e.g. beat-tracking, tempo prediction, improvisation) to be integrated in a consistent way, allowing them to be combined and/or compared systematically. In addition, the paper presents a dynamic score representation particularly suited to the demands of popular music performance by computer.

Keywords
Live Performance, Software Design, Popular Music

1. INTRODUCTION
Popular music (here regarded as music organized around a steady beat, performed live, and with sectional structure determined during performance after [1]) is an important category of music with specific characteristics that provide new challenges to computer participation in performance. Such music includes much (but not all) rock, folk, musical theatre, some jazz [1], country and contemporary (pop) church music, typically (but not always) in 3/4 or 4/4 time. The computer's role is thus no longer about providing expressive accompaniment according to a written score, or freely improvising in response to a stimulus, but instead needs to provide a more dynamic, improvised, and heterogeneous contribution to the music, that accounts for and reacts to the music played by all members of an ensemble. The nature of this genre has been previously characterized [1, 2] thus:

1. Music with steady tempo (not rigidly fixed-tempo but not as expressive in tempo as, for example, romantic period music).
2. Highly complex rhythmic patterns in place of expressive tempo variation.
3. Elaborate improvisation of individual parts leading to changes in rhythm and voicing between rehearsal and performance, and from performance to performance.
4. Tight synchronization including the possibility of playing slightly ahead of or behind the beat for expressive effect and rhythmic "feel."
5. Lack of notation detail as opposed to a full score.
6. Allowance of large structural changes during performance.

Popular music typically uses simple common practice notation, chord lists, or even memorized sections. The structure of a given piece is flexible, determined (and possibly changed) by the performers during planning, rehearsal, and performance. The music is commonly strongly sectionalized and thus can be rearranged, extended through section repetition, reordered, or cut at section boundaries. Ensembles often have mixed ability meaning that computer systems participating in an ensemble must be more tolerant of mistakes, planned substitutions of musical elements (e.g. chords), and ensemble members' absence from rehearsals.

Computer systems designed to participate in popular music ensembles (hereafter termed PM-HCMP systems) must address a range of problems including beat-tracking, tempo-prediction, score-following, ensemble listening, machine improvisation, score-management, and media synchronization. To facilitate the construction of such systems, allow easier integration of both new and existing components, and ultimately compare proposed solutions to problems and sub-problems in PM-HCMP, this paper presents a reference architecture for such systems and proposes an abstract score representation to support the kind of operations required during popular music performance. The primary contributions of this paper are therefore:

1. A reference architecture for human-computer music performance (HCMP) systems.
2. Score representations for managing the highly-structured and dynamic nature of arrangements in popular music.

The rest of this paper is organized as follows: Section 2 introduces requirements for PM-HCMP systems, describing two scenarios from different genres of popular music to illustrate the key points. Section 3 introduces the score representation proposed to support the architecture subsequently presented in Section 4. Section 5 concludes.

2. REQUIREMENTS FOR PM-HCMP

PM-HCMP systems share a range of common requirements that need to be met. These are presented below, introducing terms for the various components used later in the reference architecture and representations. Such systems need:

1. A way of representing the structure of the written score (or lead sheet or other source material) in a manner appropriate to the goal of performance (for example, elaborated measures, repeats and other notational constructs); in other words, a *static score*.
2. A simple way of representing the ordering of sections of the score without needing to recreate the static score representation in full. A simple representation is required because there is typically insufficient time to fully rewrite scores in performance scenarios, the ensembles concerned may not have the expertise to rearrange music at a fine-grained level, or indeed, some of the music may exist only as memorized blocks. This is termed the *arrangement*.
3. A way in which to transform, combine, and represent the static score and arrangement together to provide lookahead and anticipation for human and computer performers: a *dynamic score*. While the internal structure of the static score sections may remain unchanged during a performance, the dynamic score can be rewritten to account for impromptu performance decisions (e.g. repeating the chorus an additional time). The dynamic score thus begins as a representation of the future unfolding of the static score and gradually becomes a history of how that score was played. The dynamic score is analogous to the execution trace of a computer program.
4. A way in which to communicate the need for changes to the dynamic score to the performers: *cues*.
5. A way in which these representations can be communicated to a range of systems involved in supporting PM-HCMP: a *reference architecture*.

2.1 Performance Scenarios

One of the fundamental problems of PM-HCMP is in reconciling the static structure of a score with the dynamic structure of its performance. Assuming that the possibly many representations of the score have been reconciled (a non-trivial task but outside the scope of this paper to solve), the ensemble has the task of planning an arrangement for performance. This would typically involve arranging the sections of the score, determining the number of repetitions of each and so forth. The arrangement will then be rehearsed, perhaps modified, and is then ready for performance. During the performance it may be that additional repetitions are included or sections are cut at the discretion of the leader.

Much contemporary Christian music falls within the notion of popular music as defined here. One example, a commonly used song in many churches, is Beth and Matt Redman's "Blessed Be Your Name" written in 2001. There are many versions of the sheet music available for this song (for example, see [4]). A typical notated form (*static score)* would contain *Intro*|*Verse*|*Link*|*Chorus*|*Bridge*|*Coda* with repeats to indicate the sequential structure. A typical expanded pre-service arrangement (*dynamic score*) of the piece (as written by a band member following rehearsal) would be

Intro|*V1*|*V2*|*Link*|*Chorus*|*V3*|*V4*|*Link*|
Chorus|*Bridge*|*Bridge*|*Chorus*|*Chorus*|*Coda*

In preparing for rehearsal, the music leader creates (or retrieves) a static score labeling the sections as they see fit. They then produce an arrangement with the support of software that subsequently creates a dynamic score in readiness for rehearsal. During rehearsal, cueing systems prepare the computer to play and the beat acquisition systems ensure that computer parts remain in time with the other members of the band. At performance-time, the same systems operate to keep the dynamic score in time with the ensemble. If the leader decides to repeat the bridge twice more, a cue would be issued to modify the dynamic score.

As a second example, jazz performance is often based on "standards," or well-known songs specified by a melody and chord progression. A typical performance consists of an introduction, a statement of the melody, solos, and a repetition of the melody. Each soloist plays one or more "choruses." Sometimes, the last few measures are repeated (a "tag") or a special ending is played. At a gig, the band decides to play "Airegin" and adds it to a set list. In rehearsal, it was decided to have the HCMP system play guitar chords during both the melody and the piano solo and play some pre-composed string backgrounds behind the other soloists. When it comes time for the tune, the HCMP system is ready to play, listens for the count-off, and starts to play as planned. A cue is given when new soloists enter and a separate cue enables the HCMP guitar accompaniment when the piano solo starts. During the last solo, the leader decides to extend the performance by trading fours (a common structure where a musician plays four bars, the drummer plays four bars, a second musician plays four bars, etc.). Someone cues the HCMP system to "play fours," which alters the dynamic score to continue with the chord progression but to disable any music output.

Both of these examples illustrate key aspects of the HCMP problem. At performance time, it is not enough to know the static location in the score; a performer has to know the state of the current performance. For example, is this the first or second time through a repeat? The performer also needs to have a sense of intention in order to improvise successfully (e.g. building towards a final chorus). While the computer's musical material (e.g. a MIDI sequence or audio clip) can be specified statically and attached to a nominal measure, those static elements may actually map to many locations in the full-length audio of the piece (or variations in a long midi sequence). In addition, many media must be coordinated (e.g. midi, audio, electronic music display). The benefits of HCMP in both jazz and church music are the possibility of filling in for missing musicians or to augment the instrumentation of small groups.

3. Score Representation

Since the score representation is fundamental to the operation of the proposed architecture, this is presented first. The static score representation is intended to be easy to encode from a printed score or lead-sheet whilst also being amenable to arrangement and re-arrangement during performance. Since popular music arrangement typically works at the measure-level, the representations presented here operate on measures and groups of measures.

3.1 Static Score

The static score language consists of block declarations (*Decl(a)*) and terminations (*End(a)*), numbered measures (*Mx*), repeat declarations (numbered, un-numbered, dal segno), repeat terminations, and alternative ending declarations and terminations. This language allows the abstract structure of a score to be encoded without being concerned with the lower-level specification of the music material. This is beneficial for the management of multiple media during performance and for the ease of encoding and reconciliation in preparation. The

static score thus encodes the score as written at the measure level and attaches sectional labels to groups of measures.

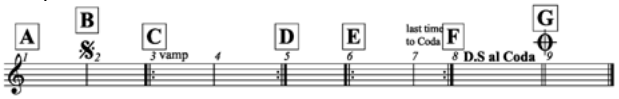
Figure 1 shows a short score fragment that will be used to illustrate the encodings proposed. The rehearsal letters indicate the designation of sections of the piece. The fragment contains a number of structural complexities including a vamp repeat (section C) to be repeated as desired by the performers, a traditional repeat and a D.S. repeat with coda. A static score encoding for this fragment is shown in Figure 1. The score encoding is relatively easy to construct quickly from reading the musical score at preparation time.

Arrangement

The arrangement uses the sectional labels declared by the static score to specify the order of the sections to be performed. This is equivalent to the musicians noting the sectional structure of the song described in Scenario 1. It allows for easy rearrangement during rehearsal and performance, simply by changing the letter ordering and regenerating the dynamic score. An example arrangement is shown in Figure 1.

3.2 Dynamic Score

The dynamic score provides a measure-level unfolding of the static score in accordance with the arrangement. Once an arrangement has been created, the measures to be played can be specified (as Mx where x is the measure number) in readiness for the render systems to schedule their data. Since it is important to be able to navigate through a piece during rehearsal (e.g. to respond to directions such as "let's go from the second time through section E"), each measure is attached to a state vector describing the sectional progress of the piece to that point.



Static Score	Arrangement	Dynamic Score
Decl(A)	A	M1[A1]
M1	B	M2[B1,A1]
End(A)	C	M3[C1_1,B1,A1]
Begin Repeat (DS)	D	M4[C1_1,B1,A1]
Decl(B)	E	M3[C1_2,B1,A1]
M2	F	M4[C1_2,B1,A1]
End(B)	B	M3[C1_3,B1,A1]
Decl(C)	C	M4[C1_3,B1,A1]
Begin Repeat (n)	D	*(CUE)*
M3	E	M5[D1,C1_3,B..]
M4	G	M6[E1,D1,C1_..]
End Repeat		M7[E1,D1,C1_..]
End(C)		M6[E2,E1,D1,..]
Decl(D)		M7[E2,E1,D1,..]
M5		M8[F1,E2,E1,..]
End(D)		M2[B2,F1,E2,..]
Decl(E)		M3[C2_1,B2,F..]
Begin Repeat (2)		M4[C2_1,B2,F..]
M6		M3[C2_2,B2,F..]
M7		M4[C2_2,B2,F..]
End Repeat		*(CUE)*
End(E)		M5[D2,C2_2,B..]
Decl Ending (DS,1)		M6[E3,D2,C2_..]
Decl(F)		M7[E3,D2,C2_..]
M8		M6[E4,D2,C2_..]
End(F)		M7[E4,D2,C2_..]
Decl Ending (DS,2)		M9[G1,E4,D2,..]
Decl(G)		
M9		
End(G)		

Figure 1: Encoding of Example Score Fragment

This captures the notion of the dynamic score being both a prescription of what is to be played and subsequently a history of what has been played. Figure 1 shows a possible dynamic score for the example fragment and arrangement shown in the figure. This is a post-performance dynamic score since pre-performance, the number of iterations of section C (the vamp section) cannot be known and it is only receipt of a cue (shown in brackets in the dynamic score) that causes the remainder of the score to be written as far as possible (until the next vamp is encountered). Unbounded repeats like this are counted during performance to support rehearsal direction (e.g. "twice through the vamp and then on"). In works without non-deterministic repeats, the entire dynamic score could be produced before performance begins.

4. Reference Architecture

Having established a score representation to support the dynamic nature of PM-HCMP, this section presents a reference architecture to capture the necessary key aspects of PM-HCMP systems. The aim of the architecture is to provide a standard organization for the components of a PM-HCMP and to give some expectations as to the type of data transmitted between them without overly constraining the design of such systems in the future. Figure 2 shows the full reference architecture that supports the key aspects of the HCMP problem.

4.1 Real-Time Components

Real-time synchronization aspects are handled by the beat and tempo tracking systems (the Beat Acquisition, Reconciliation and Prediction Modules). These should export time-stamped messages for detected pulses, meter, phase, and measures of confidence. Since there may be many of these systems, a reconciliation system is needed to filter noisy beats and decide which beat tracking source to follow on the basis of confidence and other information. This could adopt a similar approach to that outlined in [3] but accounting for the improvised nature of the music. The output of the reconciled beat data is passed to a tempo prediction system. This exports a beat-time curve to a virtual scheduler.

4.2 Abstract-Time Components

The virtual scheduler and its associated systems are concerned with the abstract time aspects of the system. The virtual scheduler retimes events scheduled on a nominal time curve by warping the curve according to the incoming tempo data from the tempo prediction system. Events are then passed to the actual scheduler for real-time scheduling. This allows the unification of all media and handles the variation of latency between the various media sources in the render system components.

The dynamic beat information (dbeat) is provided by the virtual scheduler to a structural position tracker that maintains the current score state information, mapping the dynamic score and static score and keeping the current measure count. The dbeat is a monotonically increasing beat counter and is thus inappropriate as a direct index to the static score position.

Score management is handled as described in Section 3 by the functional components in the centre of the diagram. These respond to user input (Make Static Score, Make Arrangement), and to input received from cueing systems ((Re)Make Dynamic Score).

4.2.1 Cueing Systems

Cueing systems are required to allow the computer system to react to high-level structural and synchronization changes during performance (e.g. additional repetitions of a chorus). Three types of cues are necessary:

1. *Static Score Position Cue.* This cue is necessary when synchronization with the static score is lost. Issuing it will cause the dynamic score to be re-made accordingly.
2. *Intention Cue.* This cue is needed to inform the computer of the intended direction of the current performance (e.g. exiting a vamp section or adding an additional chorus). Issuing it (e.g. using a MIDI trigger, gesture recognition or other method) will cause the future dynamic score to be remade.

3. *Voicing/Arrangement Cue.* This cue is needed to allow control over the voicing of a section (e.g. it may be desirable to prevent a particular instrumental group from playing on the first time through a repeat but allow them to play on the second). Issuing this type of cue affects only the render system to which it is issued.

4.3 Render Systems

Render systems are responsible for providing multi-media output at the appropriate time. In order to keep the detail of the specific types of media and their output separate from the abstract architecture, each render system is responsible for the management of its own data (e.g. MIDI, audio, score images). In order to link these data elements to their appropriate static score position (and thus to their appropriate scheduling as the dynamic score is played), metadata is required. A standardized format is proposed for this to relate static-score measure-numbers (and where necessary, the repeat-count, in order that the appropriate version to the dynamic score position is used) to the properties of the format concerned. This leaves render systems free to determine whether they need beat-level information or simply use the measure-level data, for example, a score display system might map a measure to image information thus: M1 → x, y, width, height, page, beat. An audio render system might represent audio at the beat level: M1 → b1:0s, b2:05s, b3…

Abstract beat-time information can thus be linked to real-time source material (to allow the correct scheduling of real-time data) while allowing the overall system to remain oblivious to the specific source formats being used. Render systems use a callback interface whereby they schedule events with the scheduling systems. These call the appropriate renderer at the scheduled time, causing synchronized real-time output of media in accordance with the dynamic score and beat tracking information. The proposed metadata representation is defined as:

metadata entry ::= measure || multi-measure || end
measure ::= (measure number, repeat-count, renderer-specific string)
multi-measure ::= (start measure number, repeat-count, count, renderer-specific string)
end ::= renderer-specific string

This generic approach allows all render systems to use the same metadata format but preserve such data as they need within the renderer-specific strings. An "end" element is required to enable renderers to terminate activities (e.g. audio playback).

5. CONCLUSIONS AND FUTURE WORK

This paper has presented a reference architecture and associated score representation for popular music human-computer music performance. The aim is to provide a standardized approach permitting easy integration of sub-systems and comparison between them. Future work will include refinements to the architecture and representations in the light of practical experience of implementing systems based on the architecture. Alternative approaches to modelling the dynamic score will be explored, for example, modelling it as a finite state machine where cue events cause state transitions. We believe that a flexible system based on this architecture will enable us to rapidly explore many variations of sensors, renderers, and interfaces as well as to integrate and share independently developed components.

6. ACKNOWLEDGMENTS

The support for this work by the EPSRC SLIM project (EP/F059442/2) and NSF Grant 0955958 is gratefully acknowledged.

7. REFERENCES

[1] Dannenberg, R.B. *New interfaces for popular music performance.* Proceedings of the 7th International Conference on New Interfaces for Musical Expression - NIME '07, ACM Press (2007), 130-135.

[2] Dannenberg, R. *Computer Coordination With Popular Music: A New Research Agenda.* Proceedings of the Eleventh Biennial Arts and Technology Symposium at Connecticut College, (2008).

[3] Grubb, L. and Dannenberg, R.B. *Automating Ensemble Performance.* Proceedings of International Computer Music Conference (ICMC-94), (1994), 63-69.

[4] Redman, B., Redman, M, *Blessed Be Your Name,* in *Here I Am to Worship*, Hal Leonard Corp, (2004), ISBN 0634079778.

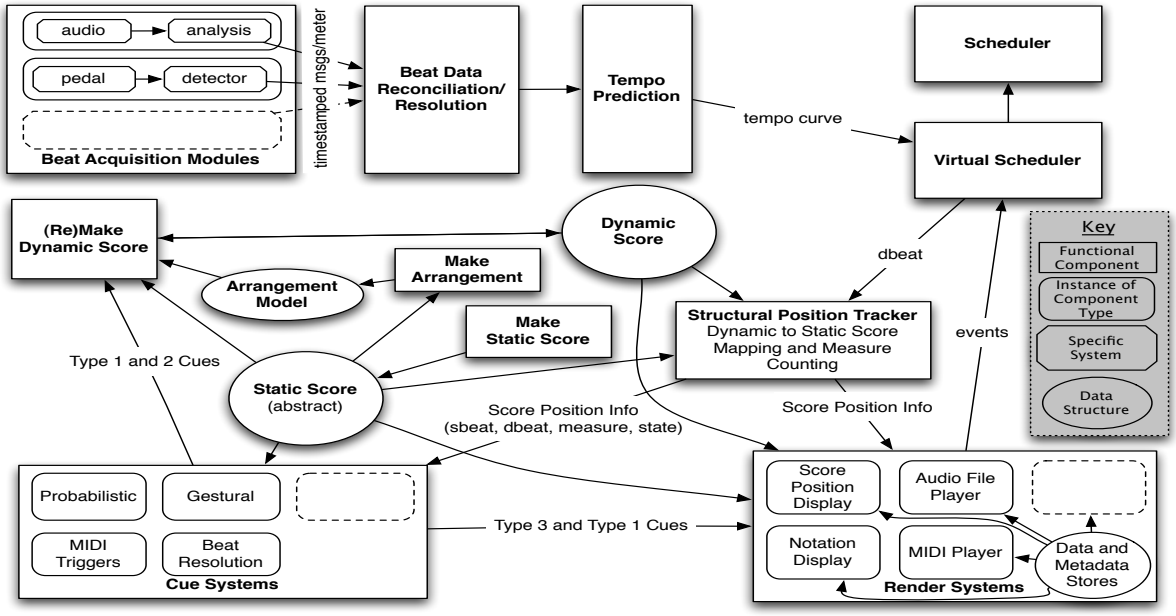


Figure 2: PM-HCMP Reference Architecture

V'OCT (Ritual): An Interactive Vocal Work for Bodycoder System and 8 Channel Spatialization

Mark A Bokowiec
University of Huddersfield
Department of Music
Huddersfield, UK
m.a.bokowiec@hud.ac.uk

ABSTRACT

V'OCT(Ritual) is a work for solo vocalist/performer and Bodycoder System, composed in residency at Dartington College of Arts (UK) Easter 2010.

This paper looks at the technical and compositional methodologies used in the realization of the work, in particular, the choices made with regard to the mapping of sensor elements to various spatialization functions. Kinaesonics will be discussed in relation to the coding of real-time one-to-one mapping of sound to gesture and its expression in terms of hardware and software design. Four forms of expressivity arising out of interactive work with the Bodycoder system will be identified. How sonic (electro-acoustic), programmed, gestural (kinaesonic) and in terms of the *V'Oct(Ritual)* vocal expressivities are constructed as pragmatic and tangible elements within the compositional practice will be discussed and the subsequent importance of collaboration with a performer will be exposed.

Keywords

Bodycoder, Kinaesonics, Expressivity, Gestural Control, Interactive Performance Mechanisms, Collaboration.

1. INTRODUCTION

In April 2010 the author undertook a 3 weeks self-directed residency at Dartington College, Devon to compose a new piece of work for solo vocalist and Bodycoder System. The departure from the last major suite of works composed for the system (*Vox Circuit Trilogy*) being that the new piece would be written for 8-channels of diffusion. The use of live and real-time performer controlled spatialization in 8-channels brought to light several challenges that needed to be addressed before the actual process of composing the new piece was started.

Before the visit various Max/MSP processing patches had been designed so that most effective use of the residency could be used to compose and rehearse the piece. Additionally an 8-channel foldback system was designed and constructed so that the performer could sensitively monitor the diffused and spatialized material.

2. BACKGROUND

The Author has been creating works for interactive performance systems since 1995, commencing with the development of *A Single Performer Controlled Mechanism for Electronic Dance/Music Theatre* - *The Navigator* (1995) [2] and *Zero in the Bone* (1997) [3] for soloist and 'Metabone'. In 1997 work began on the design and development of a flexible, wireless performer-worn sensor mechanism for interactive dance [8]. The original Bodycoder System [4] incorporated an 8-channel sensor array that used MIDI as its host protocol. Several interactive dance works resulted including *Bodycoder* (1997), *Lifting Bodies* (1999), *Zeitgeist* (1999) and *Cyborg Dreaming* (2000). A complete re-design of the Mk.1 Bodycoder took place in 2000, resulting in a doubling of the sensor channels to 16 and the use of OSC as the host protocol. It was with the creation of the performance installation *Spiral Fiction* (2002) [1] that the Mk.2 Bodycoder System was first used to control and process live vocalisation. Further experiments with solo voice lead to the composition of a suite of pieces for voice and Bodycoder system: *The Suicided Voice* (2003/7) was created during a 3 weeks self-directed residency at The Banff Centre, Canada, *Hand-to-Mouth* (2007) was composed in the EMS studios at The University of Huddersfield and *Etch* (2007) was composed, in residency on Prince Edward Island, Nova Scotia, Canada. The suite of pieces: *Vox Circuit Trilogy* (2007) had its first complete performance at The Watermans in London.

3. THE BODYCODER SYSTEM – A BRIEF DESCRIPTION

The Bodycoder System is a sensor array designed to be worn on the body of a performer. It is a performance mechanism that enables a soloist to generate, affect, manipulate and control all aspects of a multimedia performance, comprising both audio and video material. As well as movement detection sensors, the Bodycoder System also includes a number of finger mounted 'key' switches that provides the performer with the means of orchestrating and determining the nature of certain pre-defined compositional structures. The ability to work in two operational states in which sensor elements are either active and transmitting sensor data (on-line mode) or disabled with no data transmission (off-line mode) is one of the defining features of The Bodycoder System. This is a unique feature of the Bodycoder System, and is derived from the particular working practices and performance ideologies developed by the author.

To ensure maximum mobility, a radio system is employed. The sixteen channel transmitter/ PWM coder and interface unit is worn on a small belt pack and is designed to accept any combination of switched and proportional inputs. The Bodycoder System employs small resistive bend sensors,

backed with thin spring steel and enclosed with heat shrink sleeving. Each bend sensor is terminated in a small SMC screw connector that ensures that a sensor cannot be pulled out during a performance.

4. KINEASONICS

The term kinaesonic is derived from a composite of two words: 'kinaesthetic' meaning the movement principles of the body and 'sonic' meaning sound. Kinaesonics therefore refers to the one-to-one mapping of sonic effects to bodily movements and is used to describe a particular form of interactive arts practice associated with the gestural manipulation and real-time processing of electro-acoustic music [9]. The defining of the term kinaesonics was prompted by Drew Hemment's [6] description of my work with the Bodycoder System as Kinesonic: his collision of the terms Kinetic and Sonic. Kinetic implies any moving object, not specifically the human body, and this prompted me to clarify that it is the human body in relation to sound that is at the centre of my interactive practice.

5. EXPRESSIVITY

In using the word expressivity, I am not referring to an aesthetic intention that is to do with a work's reception by an audience: the indication of mood or sentiment through music. Expressivity, in terms of my work with the Bodycoder System, is a pragmatic and tangible compositional practice that is concerned with the construction and manipulation of four interactive and interrelated expressive elements: sonic (electro-acoustic), programmed, gestural (kinaesonic) and in terms of *V'Oct(Ritual),* vocal. It is the sensitive orchestration and control of the changing character of these expressive elements and the choices made with regard to the manner of their interaction and influence on each other that defines the practice and ultimately the individual nature of the resulting works. With respect to the Bodycoder System, and particularly in relation to *V'Oct(Ritual),* expressivity can be sub-divided into four principle forms.

- Gestural = G
- Sonic = S
- Vocal = V
- Programmed = P

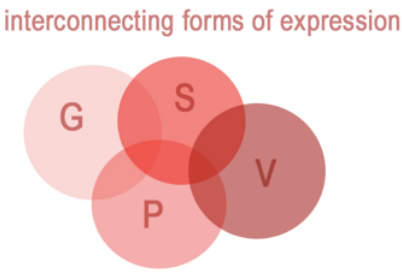


Figure 1. Four principle forms of expressivity

The four forms of expressivity are inter-related and interact with each other in various ways and degrees. An awareness of the interconnectivity of principle forms of expressivity, their interaction and influence on each other, shapes the compositional, development and rehearsal processes.

5.1 Gestural (Kinaesonic)

Gestural (kinaesonic) expressivity refers to the physical movements made by the performer. Gestural expressivity is intimately linked to programmed expressivity through scaling and mapping within Max/MSP that models the kinaesonic relationship between sound processing and physical gesture. Gestures and their location on the body are largely dictated by the performance demands of the composition and ease of articulation. Real-time gestural control of live electro-acoustic processing requires a high degree of physical skill, musicality and aural awareness. The flexibility of the Bodycoder System's hardware, protocols and functionality means that gestural expressivity can be uniquely configured for a range of physicality that corresponds to different types of kinaesonic expressions from moment to moment within a piece.

5.2 Sonic

Sonic expression is concerned with the way in which sound subjected to processing and often re-processing[1] evolves over time and can be layered to create dynamic and dimensional soundscapes. Sonic expressivity in terms of my own compositional practice is founded on this notion of evolution and duration. Such evolutions are considered physical/organic in that they are programmed with a quality of movement (transformation) within the larger sonic landscape[2]. *V'Oct(Ritual)* uses a combination of granularization, compression looping and filtering to create multiphonic layers of sound, portions of which may be subjected to live gestural articulation. Equally, such transformations may operate as separate entities that are not subjected to any form of additional gestural articulation by the performer. In this case their programmed, shaped and automated evolution alone and not their live/gestural articulation is considered expressive. Therefore sonic expression can be modelled entirely within the DSP processing through the programming of variables to create automated sonic events and/or expression that can be shaped (controlled and articulated) through gesture (kineasonics): the gesture of the performer defining the scale and time-frame of sonic transformation. In both cases the nature of the sonic transformation is programmed and scored.

5.3 Programmed

Qualities of sonic expression are modelled in the Max/MSP environment through the use of mapping and scaling processes to translate degrees of physical gesture to control electro-acoustic processes. Expressivity is tuned through the mapping of different ranges of audio and/or visual processing to, for instance, the bend of the arm, wrist etc. Various mapping ratios, for example the proportion of an arm movement to a particular range of sonic manipulation, produce specific physical expressivity. Different scaling ratios vary from sensor to sensor and can be changed from moment to moment within a piece. The real-time expressivity of kinaesonic actions is

[1] This might include timbral and textural development, transformation through fragmentation, the use of randomisation and chance processing, transformation through pitch change, spatialisation and evolution through the use of various mixing and fading techniques.

[2] This idea has some correspondence with the notion of *gestural sonorous objects* further explored in Von Nort, D. (2009) [7].

established through these mapping and scaling choices during the rehearsal process according to various performance preferences including the ease/difficulty of physical execution and the quality of control required.

5.4 Vocal

The expressivity of the acoustic voice is important not only with respect to its unprocessed presence within the sonic landscape, as something of a soloist, but more crucially in the manner in which it interacts with live processing. In *V'Oct(Ritual)* the timbre, pitch and energy of the acoustic voice is used to enliven, activate and articulate certain electro-acoustic processes. A key part of the development of *V'Oct(Ritual)* was concerned with identifying the qualities of acoustic vocal input that resulted in sonically rich interactions. The same concerns informed the choice of phrasing, melody construction, the quality of accents and the use of natural forms of vocal filtering - executed by changing the shape of the mouth and the muscular use of the throat and the larynx: generically known as extended vocal techniques.

6. V'OCT(RITUAL)

6.1 Protocols and Functions

The interface used for *V'OCT(Ritual)* employs twelve switched inputs, four finger switches on a right hand data glove provide individual sensor activation and deactivation, i.e. facilitating on-line and off-line modes of operation. Eight finger switches mounted on the left hand glove provide utility functions such as Max/MSP patch/preset selection and granular sampling and recording. Bend sensors are located, one on each elbow and one on each wrist, the mapping and programmed expressivity (sensor scaling) of each sensor element can be changed during the course of a piece of work.

As in all previous works created for the Bodycoder System the performer is required to control all aspects of the performance with no off-stage intervention from the mixing desk/computer system. In *V'OCT(Ritual)* this includes patch/preset navigation, initiation of granular sampling, compression recording, activation, routing and control of filter and pitch processes and initiation and gestural (kinaesonic) control of various spatialization routines. In terms of spatialization the activation of either wrist sensor, via the right hand data glove, routes the outputs of the selected granulator to one of three spatialization processors. In this way the individual granulator output phases can either move repeatedly between output channels or be gesturally spatialized by the wrist sensors.

6.2 Max/MSP Design

The Max/MSP design for *V'OCT(Ritual)* is based around the principles of granular sampling and compression looping. The main DSP patcher includes two 8-channel compression loopers (each including an 8-channel low-pass filter), two 8-channel granulators and three 8-channel spatializers. The first compression looping patcher consists of eight recording/ playback buffers, the size of each buffer variably pre-set via message boxes, stored in patch presets, that are recalled by the performer. This patcher is designed so that with the onset of a recording command, generated by the activation of a dedicated finger switch, each buffer is sequentially loaded with new vocal material. The output of each buffer is routed to individual output channels. The second compression looper operates by recording into pairs of buffers designated front narrow, front wide, rear wide and rear narrow. In this case the recording of the live vocal signal is sequentially loaded into the front pair of recording buffers through to the rear pair of recording buffers. The two granulators each output eight, equally-spaced, grain phases that are either connected to a discrete output channel or mixed and fed to one of the three spatialization processors. A master patcher handles all signal routing and processing patcher activation and muting. The master patcher also includes a sensor sub-patcher, a preset messaging patcher and a TouchOSC patcher. The TouchOSC patcher sends patch and recording feedback cues to the performer enabling visual monitoring on an Apple iPod Touch using the TouchOSC application.

6.3 Mapping Strategies for Spatialization

One unusual feature of *V'Oct(Ritual)* is the combination of automated (programmed) and live (performer controlled) spatialization with the performer deciding when it is appropriate to take control of sonic diffusion and the appropriate mode of spatialization.

Automated spatialization operates in two modes, each mode unique to each of the two different granulator abstractions. The first mode operates by randomly positioning each granulator phase signal across individual output channels. The width and speed of panning is pre-set and stored for recall by the performer. The second mode moves the granulator phases through a sequence of preset trajectories that are again recalled by the performer as part of the patch preset recall sequence.

Gesturally controlled spatialization operates in three modes. The first mode is enabled by the simultaneous activation of both wrist sensors. This effectively mixes the eight grain phases of the active granulator into a pair of channels, each comprised of four grain phases. These mixed granular pairs are routed so that one channel can be gesturally panned between the front and rear channels (right hand side) and the front and rear channels (left hand side) using the sensor elements located on the right and left wrist respectively. The second spatialization mode is enabled by the activation of an individual wrist sensor that effectively routes a mix of all grain phases to two rotational spatializers, the right wrist controlling a panning in an anticlockwise direction and the left wrist controlling a panning in a clockwise direction. The remaining spatializer is selected by the operation of a dedicated finger switch. Once this switch has been detected a mix of all eight granulator phases is routed to a triggered panner. Subsequent detection of this finger switch pans the combined signal from its current location to a randomly selected output channel. The duration of each pan trajectory is dynamically controllable by the right wrist sensor, operating in a range of between 0 and 2500mS.

6.4 Eight Channel Monitoring System

In designing a piece for interactive, performer controlled spatialization it is of paramount importance that the performer can monitor the live and processed vocalisations without having to be situated in the 'sweet spot' of an auditorium. To achieve an intimate and sensitive level of control a custom 8-channel monitor array was designed and constructed using relatively cheap, active computer monitors. Each pair of *Bose Soundsticks II* employs a floor mounted sub bass unit that also houses the amplifier circuitry. The level of signal sent to each sub bass can be controlled via a control on each unit that allows a balance to be set up between each pair of mid/high drivers and the sub bass driver. The mid/high range speakers were mounted on fabricated brackets mounted on round-base microphone stands, see Figure 2. Each Bose mid/high unit incorporates four individual drivers in a vertical housing that transmits a highly focused sound source that is ideal for

multichannel, close-field monitoring. 8-channels of audio is transmitted from the mix position via an ADAT optical link utilizing an optical line driver/receiver to ensure signal integrity. It is important that the performer has independent control over the signal sent to the monitor array. To achieve this a custom MIDI foot pedal is employed together with a MIDI line driver/receiver sending a simple MIDI controller signal to the computer system.

Figure 2. *V'Oct(Ritual)* in Rehearsal

7. COMPOSITIONAL PROCESSES

Working collaboratively with a performer is not only a conscious artistic choice but one that is necessitated by the real-time and interactive nature of the work. In terms of *V'OCT(RITUAL),* the acoustic vocalisations of the performer form the raw input material of the piece – this too is difficult to simulate without the presence of the performer.

Programmed expressivity such as sensor scaling, mapping and response composed within the Max/MSP software, also impacts upon the physicality (gestural expressivity) of the performer, it is therefore necessary that the performer participates in decisions that prescribe their physicality.

Because of the level of real-time control and responsibilities for both initiation and navigation of the Max/MSP environment as part of the realisation of the live performance, it is necessary that the performer is completely cogent with the larger hardware and software architecture of the piece. This knowledge is established through the compositional /development and rehearsal phases of a piece.

The development and learning of the acoustic vocal score, the internalising of the gestural kinaesonic score, and an understanding of the larger architecture of a piece is established over periods of intensive rehearsal.

The performer's collaborative input and their intimate knowledge of the architecture of a work is a defining characteristic of the practice. This knowledge affords the performer both security within the live performance /composition and a level of autonomy that excludes the need for outside interventions from the mixing desk. This produces a truer level of virtuosity, not simply in terms of quality of gestural and vocal expressivity, but also in terms of self-determined control within the pre-composed structures.

8. CONCLUSION

Advancing into the area of performer controlled spatialization is a new development in my practice which poses some interesting aesthetic and technical problems. It is an area of interactive and electro-acoustic music practice that for a number of years has been generating debate with regard to the authority of the performer over the diffusion of their own instrument. Simon Emmerson suggests "we might consider giving the performer some say over what happens in projecting field information. This would complete our idealized control revolution returning considerably more power to the performer than current systems allow" [5]. In terms of my own future practice with the Bodycoder System my chief concerns are how to integrate spatialization into the compositional integrity of works in terms of sonic and programmed expressivities. Also how gestural spatialization is executed in such a way that it is not seen as merely demonstrative. Gestural spatialization also adds to the control responsibilities of the performer and it is expected that there will be a range of skills and particular perceptions that will need to be more clearly identified and refined. This may change established patterns of practice and will inevitably add another dimension to the collaborative process.

9. REFERENCES

[1] Bokowiec, M.A., and Wilson-Bokowiec, J. (2003). 'Spiral Fiction' in *Organised Sound*. Cambridge University Press. (8) 8, pp. 279-287.

[2] Bromwich, M. (1995). 'A Single Performer Controlled Interface For Electronic Dance/Music Theatre' in *The Proceedings of the International Computer Music Conference*. International Computer Music Association, San Francisco, CA. pp. 491-492.

[3] Bromwich, M. (1997). 'The Metabone - An Interactive Sensory Control Mechanism For Virtuoso Trombone' in *The Proceedings of the International Computer Music Conference*. International Computer Music Association, San Francisco, CA. pp. 473–475.

[4] Bromwich M., and Wilson J. (1998), 'Bodycoder: a Sensor Suit and Vocal Performance Mechanism for Real-time Performance' in *Proceedings of* the *International Computer Music Conference.* International Computer Music Association, San Francisco, CA. pp. 292-295.

[5] Emmerson, S (2007) *Living Electronic Music.* Ashgate Publishing Limited. Hampshire & Burlington TV. p. 96

[6] Hemment, D. (1998). 'Bodycoder and the Music of Movement' in *Mute* magazine. Issue 10, pp. 34-39.

[7] Von Nort, D. (2009). 'Instrumental Listening: Sonic Gesture as Design Principle' in *Organised Sound*. Cambridge University Press. (14) 2, pp. 177-187.

[8] Wilson-Bokowiec, J., and Bromwich, M. (2000). '*Lifting Bodies*: Interactive Dance – Finding New Methodologies in the Motifs Prompted by New Technology – a Critique and Progress Report with Particular Reference to the Bodycoder System' in *Organised Sound* (5) 1 Cambridge University Press, pp. 9-16.

[9] Wilson-Bokowiec, J., and Bokowiec, M. (2006). 'Kinaesonics: 'The Intertwining Relationship of Body and Sound' in *Contemporary Music Review. Special Issue: Bodily Instruments and Instrumental Bodies.* London, Routledge Taylor & Francis Group, Volume 25 Number 1+2, 2006, pp. 47-58.

First Person Shooters as Collaborative Multiprocess Instruments

Florent Berthaut
LaBRI - SCRIME
florent@hitmuri.net

Haruhiro Katayose
Kwansei Gakuin University
katayose@kwansei.ac.jp

Hironori Wakama
Kwansei Gakuin University
cuh79618@kwansei.ac.jp

Naoyuki Totani
Kwansei Gakuin University
naoyuki.totani@kwansei.ac.jp

Yuichi Sato
Kwansei Gakuin University
bih62181@kwansei.ac.jp

ABSTRACT

First Person Shooters are among the most played computer video games. They combine navigation, interaction and collaboration in 3D virtual environments using simple input devices, i.e. mouse and keyboard. In this paper, we study the possibilities brought by these games for musical interaction. We present the Couacs, a collaborative multiprocess instrument which relies on interaction techniques used in FPS together with new techniques adding the expressiveness required for musical interaction. In particular, the *Faders For All* game mode allows musicians to perform pattern-based electronic compositions.

Keywords

the couacs, fps, first person shooters, collaborative, 3D interaction, multiprocess instrument

1. INTRODUCTION

In the past years, success of musical video games has grown quickly, with games such as Rock Band, Guitar Hero, Wii Music and so on. Some of these games use or imitate musical instruments, adding evaluation of players performances. These games rely on musical interaction techniques. We believe that, in the same way, musical interaction can benefit from techniques developed for video games using devices such as keyboards, mice, gamepads, joysticks. In fact, players may develop specific skills, as described in [15], and enhance some abilities that musical instruments may build on. An especially interesting genre of video game are First Person Shooters (FPS).

These games feature rich 3D virtual environments that can be used to facilitate control and visualization of multiple sound processes.

In this paper we discuss the possibilities but also the issues brought by First Person Shooters for musical interaction. Then we present the Couacs, a collaborative multiprocess instrument which focuses on interaction between players/sound processes. A picture of three musicians playing the Couacs can be seen on figure 1, and a presentation video is available [1]. In particular, we propose and evaluate a game mode called *Faders for All*.

2. RELATED WORK

Some musical video games rely on performance evaluation. For example in Rock Band [2], the goal is to play as much correct notes as possible according to a score. They are usually played with input devices that imitate musical instruments. Other games such as Rez [3] rely on traditional gaming devices and consist in triggering predefined musical events. In these games, however, musical freedom, as defined by Jordà [14], and expressiveness are limited as explained by Marczak et al. [16]. Interestingly, although games such as *Rock band* are multiplayer, there is almost no interaction between players. Each player focuses on his own actions.

On the other hand, in the past years several laptop orchestras were created, such as the Standford Laptop Orchestra [18]. They make use of common devices used in laptop music [6], i.e. mouse and keyboard. In addition they enable interaction and musical dialog between several musicians.

Much research has been done on using common input devices for musical interaction, for example by Zadel et al. [20] or Fiebrink et al. [8]. Often 2D graphical interfaces are used in conjunction with these devices.

Some new instruments rely on first person navigation in a 3D environment. For example, the musical piece *La ménagerie imaginaire*, built upon the research done by Wozniewski et al. [19], allows musicians to apply effects on their acoustic instruments by navigating in a virtual environment.

Finally Hamilton developed several instruments and systems based on FPS. Maps and legends [9] is an instrument which makes use of navigation and sound spatialization, by superimposing the virtual environment and the concert room. Q3osc[10] is a modification of a game engine which outputs OpenSoundControl messages for every game parameter. Weapons projectiles may then be associated with sounds triggered on collisions with the environment.

Rather than sound spatialization, we are interested in interaction between musicians and control of multiple sound processes in the 3D environment. We want to take advantage of skills developed by FPS players to enable expressive musical interaction.

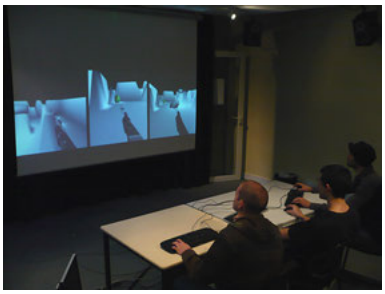


Figure 1: Three musicians playing the Couacs in *Faders For All* mode

3. FIRST PERSON SHOOTERS

3.1 Overview

In First Person Shooters, players control 3D avatars and perceive the 3D virtual environment through their eyes with a first person

[1] http://www.vimeo.com/19347468
[2] http://www.rockband.com/international
[3] http://www.sonicteam.com/rez

NIME'11, 30 May–1 June 2011, Oslo, Norway.

perspective. The gameplay consists in navigating in the environment, usually using the keyboard, and shooting at other players with different weapons. The mouse is used to aim, change weapon and shoot. When a player gets killed, he starts again in another spot of the environment. Items such as health, shield, weapons, invincibility, or speed can be found in the environment and picked up by the avatars, usually simply by walking through them. In multiplayer FPS, the goal depends on the chosen game mode. For example, in a *Free For All* (FFA) game, the goal is to have more "frags", i.e. kills, than other players at the end of the game. In a *Capture the Flag* game, the goal is to grab the flag from the other team's camp and bring it back to one's camp. Different virtual environments, called *Maps*, are used, such as indoor maps with several rooms, terrains with hills and trees, platforms in space and so on.

3.2 Musical FPS

In this section, we study the possibilities provided by the use of First Person Shooters as musical instruments, but also the issues that it raises.

Interaction

First of all, interaction techniques developed for FPS provide several control dimensions. Navigation in the environment can be used for example as several continuous parameters with the absolute position and rotation. But it can also be interesting as a discrete parameter with movements states such as crouching, jumping, running and so on. Items can be used for discrete modulations of sound parameters, for example it may affect several sound processes at the same time. Weapons and shooting have several parameters such as weapon type or weapon mode. Finally, FPS make use of bimanuality, with one hand handling large movements using the keyboard while the other hand performs more accurate movements to aim and shoot.

On the other hand, input devices used in FPS are common and affordable but they often restrict musical freedom and expressiveness. A mouse button only outputs a 1 bit value, so that one can only control the rhythm of clicks but not their velocity. This prevents players from correctly performing instantaneous excitation gestures as described by Cadoz [5]. On the other hand, graphical actions, such as translations and rotations, provide a good spatial resolution but they can not be done with as much temporal accuracy as mouse clicks, due for example to the latency of graphical rendering. Therefore we have to provide additional degrees of freedom for gestures performed using the mouse and keyboard.

Collaboration

Shoot, touch and other interactions between avatars can be used as musical metaphors for various musical interactions between sound processes. Game modes may be a way of switching from one metaphor to another, and therefore between collaboration modes.

Visualization

As stated by Jordà [13], graphical interfaces allow for efficient interaction with several sound processes, by giving informations on their state and parameters, and by facilitating access to sound processes. FPS allow to visualize sound parameters using different 3D graphical elements. First of all, the environment may represent musical structures. Avatars may be used to display individual or combined sound processes. Projectiles fired by weapons also have several parameters that can be used to control sound parameters. Finally, graphical effects such as shading can be applied to the whole environment to represent effects applied to all sound processes or global mood of a song.

On the other hand, we need to correctly choose how to represent these sound processes in the environment in order to easily identify these processes and visualize their parameters.

Immersion

FPS provide a good sensory immersion [7] compared with other video games. This immersion is especially interesting for the players, as it may improve the implication in the instrument. But it can also be interesting for the audience as it may ease the understanding of players' actions.

Accessibility/Spreading

Compared with other 3D instruments, FPS-based instruments make use of simple and common input devices such as keyboards and mice, instead of six degrees-of-freedom tracking systems. This may facilitate the spreading of such instruments among gamers and laptop musicians, as they usually already have all the needed hardware. Furthermore, communities built around these games may also help improving these instruments by creating new game modes, new maps and organizing Local Area Network (LAN) Parties or Tournaments and even Concerts. This may partly solve the problem of most new instruments which are never played again after the first paper/concert.

Learning

Learning and gaining expertise is an important issue for new instruments. Existing FPS tournaments prove that players can improve their skills. Eventually, some players become virtuoso by mastering all game techniques and improving their accuracy and reactivity.

Game or Instrument

A final question is the balance between gaming and playing music. How can we use some game actions for musical control without disturbing other game actions not connected to sound, and vice-versa ? Will gamers/musicians try to learn how the instrument works and how they can produce specific musical results or will they only play without paying attention to the generated music ? Should these instruments have a goal like a video game or not ?

4. THE COUACS

In this section, we present the Couacs, a collaborative multiprocess instrument based on First Person Shooters. This instrument allows us to experiment the adaptation of interaction and collaboration techniques used in FPS to musical interaction. It uses Irrlicht[4] for graphical rendering, Jack[5] for sound rendering and libxtract [4] for audio features extraction.

4.1 General approach

In the Couacs, each musician controls a 3D avatar associated to one or several sound processes. Actions and characteristics of the avatars modify the sound processes, and in return the aspect of the avatars reflects properties of the sound processes. The Couacs enables the use of several mice and keyboards simultaneously, so that several musicians can play with the same computer in split-screen mode. Each game mode may be a totally different instrument with different sound processes and mappings. For now only the *Free For All* mode, renamed *Faders For All*, has been implemented. In section 4.2, we present and evaluate this game mode regarding the possibilities and issues described in section 3.2.

4.2 Faders For All Game Mode

The first game mode implemented is called Faders For All. In this mode, each avatar is associated to a different sound process, i.e. instrument, composed of a base pattern with several sound samples and several audio effects. When an avatar shoots, it triggers a variation of its associated sound processes. If one avatar shoots another avatar and hits it, the triggered variation and the effects are imposed to the sound process of the player that has been hit. Each time an avatar is hit, the volume of its associated sound process is reduced. It can be recovered by grabbing health items. Therefore, the musical result oscillates between base patterns, mix between sound processes, solo breaks and joint breaks. This mode is aimed at electronic music performances. Pattern-based compositions can be translated into songs defined in files with an XML syntax, containing instruments definitions and patterns. Each instrument is then controlled by a different musician and may interact with other instruments.

4.2.1 Interaction / Expressiveness

As explained in section 3.2, in order to be able to perform expressive instrumental gestures, and especially instantaneous excitation

[4] http://irrlicht.sourceforge.net/

[5] http://jackaudio.org/

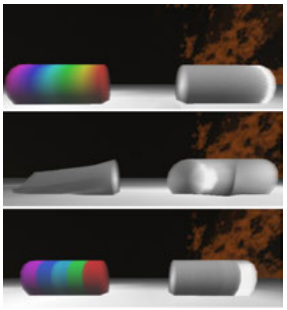


Figure 2: Tunnels (left to right, top to bottom): Continuous Hue, Continuous Density, Continuous Distortion, Continuous Rotation, Discrete Hue (5 values), Discrete Density (5 values).

gestures as defined by Cadoz [5], we need to extend the gestures done using mice and keyboards. Therefore we use the interaction techniques which were developed for an input device called *Pivert* [2], in particular percussion gestures. These gestures were designed to perform instrumental gestures with several parameters, such as velocity, direction and duration. For example, high-level *Flam* gestures are composed of two successive low-level *Hit* gestures, here button clicks, done with different fingers. *Roll* gestures are composed of three *Hit*. Instead of having only two 1-bit gestures on the mouse, we obtain two gestures, i.e. Flam and Roll, both with a 1 bit direction parameter and a duration parameter encoded on at least 7 bits (depending on the accuracy of time measurement). Therefore gesture duration can be used to replace velocity that would be provided by a pressure sensor. With these gestures, one may perform temporally accurate and expressive instantaneous excitation or modulation gestures, of course with some training

In addition to gestures done using the mouse, avatars movements can be used to control sound parameters. But as explained in section 3.2, we believe that using these parameters should not force players to move to fixed positions, i.e. they should be able to control them anywhere in the environment. This is why the Couacs relies on movement states instead of absolute position and rotation of the avatars. We define ten movement states which reflect movements with increasing dynamics, i.e. *Crouch, Stand, Crouch_Walk, Backward, Strafe, Run, Jump, Jump_Back*, *Jump_Strafe*, *Jump_Forward*. This gives us a 10 values discrete parameter that can be used in conjunction with mouse gestures for example to provide an additional parameter to excitation gestures.

Along with the fast discrete gestures performed with the mouse and the keyboard, the Couacs allows for graphical modulations of the sound processes parameters, using 3D graphical tools called *Tunnels* [1]. Players modify avatars parameters, such as color or transparency, and therefore their sound processes by moving through these tools. On the contrary to traditional graphical sliders, *Tunnels* may control one or several parameters with several discrete or continuous scales, as depicted on figure 2.

Usual 3D graphical items, such as portals, health and special abilities, are also used to enrich interactions with the environment and provide other musical possibilities, such as switching from part of a song to another.

4.2.2 Collaboration

Collaboration in the *Faders For All* game mode relies on the shooting metaphor and allows players to modify other players sound processes.

When an avatar shoots another one, it imposes its sound process variation to the other sound process, which means that for a short period, they will play notes simultaneously. At the same time, avatar parameters and their associated audio effects parameters are copied to the avatar that has been shot. Therefore, players try to shoot other players to influence both their pattern and their audio effects. Usually this leads to short musical dialogs, but also transitions between atmospheres since the audio effects tend to propagate among players, e.g. a player that has been shot shoots another player.

4.2.3 Learning / Mappings

Each weapon corresponds to a different set of mappings between input, game and sound parameters. Therefore, weapons with simple one-to-one mappings can be used by beginners while more advanced musicians can use many-to-many mappings as described by Hunt and Kirk [12]. Weapon selection thus modify the *Expertise* needed for the instrument as called by Wanderley et al. [17], along with the *Musical Freedom* described by Birnbaum et al. [3]. As in most FPS, weapons can be selected using the scroll wheel of the mouse. For now, four weapons have been implemented: *Velocity*, *Pitch*, *Repeat*, and *Multi*. When selecting a weapon, only the projectile, which hangs at the end of the weapon, changes as it can be seen on figure 3. Each projectile type represents a different control effect applied to base patterns.

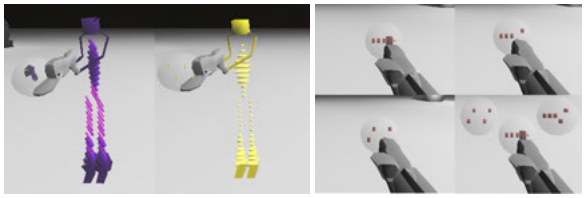


Figure 3: Left: Avatar and projectile with different aspects reflecting the audio effects applied to the sound process. Right: Projectiles for the Velocity Weapon/Effect, Pitch Weapon/Effect, Repeat Weapon/Effect. The *Multi* Weapon combines the three projectiles.

For the *Velocity*, *Pitch* and *Repeat* weapons, one triggers variations, i.e. activations of corresponding control effects, using *Hit* gestures done on the mouse. Effects values are mapped to movement dynamics, i.e. movement state ranging from *Crouch* to *Jump Forward*. In addition, the choice of the finger for hit gestures, i.e. mouse clicks, controls effects spreading.

The *Multi* weapon allows one to control the three control effects almost simultaneously since the movement state sets the projectile type and thus the triggered control effect. When standing or moving backward, this weapon triggers the velocity effect which modifies velocities of pattern notes. When strafing, it triggers the pitch effect which modifies pitches of pattern notes. When moving forward, it triggers the repeat effect which repeats pattern notes. If these movements are done while jumping, the duration of the effect is increased. The *Multi* weapon also mutes the sound process when there is no movement at all. Variations are triggered using *Flam* gestures, with gesture duration controlling the control effect value and gesture direction controlling the spread parameter. The *Multi* weapon requires more expertise since it uses more complex mappings and gestures. On the other hand it offers more musical freedom.

During an informal study, users confirmed that in order to learn how to play in the *Faders For All* mode, one must start with simple patterns, e.g. a single note, and the first weapon in order to understand which sound process they control, how to apply effects with tunnels and how to interact with other players. Then they may switch to other weapons to gain musical control. Finally, one of the users commented that it would be interesting for expert musicians to have an additional weapon allowing them to trigger the notes of patterns themselves.

4.2.4 Visualization

For each sound process, Bark coefficients of the spectrums of all sound samples are added and used to set the shape of the associated avatar by scaling cubes composing its body, from lowest frequencies on its feet to highest frequencies near its head. Loudness is also analysed in real-time and modifies the scale of the avatar. This combination of static and dynamic analysis and visualization allows players to identify other players sound processes and follow their activity.

Graphical parameters of avatars and projectiles can be modified by moving through the *Tunnels*, or by being shot. Color Hue, Rotation, Shape Distortion and Density parameters are mapped to

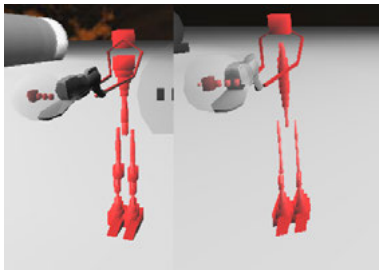


Figure 4: Avatars corresponding to a synth pattern with high and low frequencies (left) and to a drums (kick, snare, hi-hat) pattern (right).

audio effects of sound process. These visual features are combined on models to visualize sound processes parameters, taking inspiration from research done in the information visualization field by Healey [11]. They can be reset to the default values using a *Three-strike Roll* "Right-Left-Right" gesture with the mouse.

When a player is hit, the opacity of his avatar decreases. The opacity is mapped to the sound process volume which also decreases. When the sound process is almost silent, the avatar is almost completely transparent. That gives an advantage to the player. The player then needs to pick up an Health Item to restore the opacity and volume.

4.2.5 Game or Instrument

Goals of the original game mode, i.e. to kill the other players and to avoid getting killed, are preserved. In fact, since sound process volumes are associated to players health players don't want their volume to get reduced. They also try to shoot other players in order to influence their sound processes and therefore the global musical result.

5. CONCLUSION

First person shooters are characterized by highly dynamic gestures, expert interaction techniques, visualization and collaboration possibilities, and strong communities. Digital musical instruments may build on these advantages to provide new expressive interfaces while solving issues peculiar to musical interaction.

The Couacs is a collaborative multiprocess instrument based on FPS. It makes use of gaming interaction techniques and adds techniques such as *Tunnels* and percussion gestures to improve expressiveness of mouse gestures. It allows for the visualization of sound processes parameters and audio perceptual features using 3D avatars, weapons or the environment. In the *Faders For All* mode, a shooting metaphor allows for musical dialog between players.

The first perspective is the evaluation of the *Faders for all* game mode, with both musicians and gamers, in terms of musical control, learning curve, collaboration and visualization.

In order to explore new collaboration possibilities, we are working on other game modes. In particular, in the *Capture The Fader* (originally Capture the Flag) game mode, there are two teams, with base camps on each end of the environment, associated with two synchronized songs and a 3D flag acting as a crossfader on a dj mixer. The following game mode will be the *Rhythm Chase* mode (originally Rabbit Chase), in which one player holds a pattern which the other players complete with occurrences of their sound by shooting him until he drops the completely filled pattern.

6. ACKNOWLEDGMENTS

This work was made possible by a grant from the Centre National de la Recherche Scientifique (CNRS) and the Japan Science and Technology Agency (JST). Florent Berthaut would also like to thank Pascal Guitton for the opportunity and Professor Katayose and his team at Kwansei Gakuin University.

7. REFERENCES

[1] F. Berthaut, M. Desainte-Catherine, and M. Hachet. Interaction with the 3d reactive widgets for musical performance. In *Proceedings of Brazilian Symposium on Computer Music (SBCM09)*, 2009.

[2] F. Berthaut, M. Hachet, and M. Desainte-Catherine. Piivert: Percussion-based interaction for immersive virtual environments. In *Proceedings of the IEEE Symposium on 3D User Interfaces*, 2010.

[3] D. Birnbaum, R. Fiebrink, J. Malloch, and M. M. Wanderley. Towards a dimension space for musical devices. In *NIME '05: Proceedings of the 2005 conference on New interfaces for musical expression*, pages 192–195, Singapore, 2005. National University of Singapore.

[4] J. Bullock. Libxtract: A lightweight library for audio feature extraction. In *Proceedings of the International Computer Music Conference*, 2007.

[5] C. Cadoz. *Musique, geste, technologie*. Éditions Parenthèses, 1999.

[6] K. Cascone. Laptop music - counterfeiting aura in the age of infinite reproduction. *Parachute, issue 107*, 2002.

[7] L. Ermi and F. Mäyrä. Fundamental components of the gameplay experience: Analysing immersion. In *DiGRA conference Changing views: worlds in play*, 2005.

[8] R. Fiebrink, G. Wang, and P. R. Cook. Don't forget the laptop: using native input capabilities for expressive musical control. In *Proceedings of the 7th international conference on New interfaces for musical expression*, NIME '07, pages 164–167, New York, NY, USA, 2007. ACM.

[9] R. Hamilton. Maps and legends: Designing fps-based interfaces for multi-user composition, improvisation and immersive performance. *Computer Music Modeling and Retrieval. Sense of Sounds: 4th International Symposium, CMMR 2007, Copenhagen, Denmark, August 27-31, 2007. Revised Papers*, 2008.

[10] R. Hamilton. q3osc: or how i learned to stop worrying and love the game. In *Proceedings of the International Computer Music Association Conference*, 2008.

[11] C. G. Healey. Building a perceptual visualisation architecture, 2000.

[12] A. Hunt and R. Kirk. Mapping strategies for musical performance. *Trends in Gestural Control of Music*, pages 231–258, 2000.

[13] S. Jordà. Interactive music systems for everyone: exploring visual feedback as a way for creating more intuitive, efficient and learnable instruments. In *Proceedings of the Stockholm Music Acoustics Conference (SMAC03)*, 2003.

[14] S. Jordà. *Crafting musical computers for new musics' performance and improvisation*. PhD thesis, Universitat Pompeu Fabra, 2005.

[15] P. Kearney. Cognitive callisthenics: Do fps computer games enhance the player's cognitive abilities? In *Proceeding of the International DiGRA Conference*, 2005.

[16] R. Marczak, M. Robine, M. Desainte-Catherine, A. Allombert, P. Hanna, and G. Kurtag. Enhancing expressive and technical performance in musical video games. In *Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference*, 2009.

[17] M. Wanderley, N. Orio, and N. Schnell. Towards an analysis of interaction in sound generating systems. In *ISEA2000 Conference Proceedings*, 2000.

[18] G. Wang, N. Bryan, J. Oh, and R. Hamilton. Stanford laptop orchestra(slork). In *Proceedings of the International Computer Music Conference*, pages 505–508, 2009.

[19] M. Wozniewski, Z. Settel, and J. Cooperstock. A spatial interface for audio and music production. In *Proceedings of the International Conference on Digital Audio Effects (DAFx), 2006*, 2006.

[20] M. Zadel and G. Scavone. Different strokes: a prototype software system for laptop performance and improvisation. In *Proceedings of the 2006 conference on New interfaces for musical expression*, NIME '06, pages 168–171, Paris, France, France, 2006.

Studying Interdependencies in Music Performance: An Interactive Tool

Tilo Hähnel, Axel Berndt
Otto-von-Guericke Universität Magdeburg
Universitätsplatz 2
39106 Magdeburg
{tilo,aberndt}@isg.cs.uni-magdeburg.de

ABSTRACT

Musicians tend to model different performance parameters intuitively and listeners seem to perceive them, to a certain degree, unconsciously. This is a problem for the development of synthetic performance models, for they are built upon detailed assumptions of several parameters like timing, loudness, and duration—and of interdependencies as well. This paper describes an interactive performance synthesis tool, which allows to analyse listener's preferences of multiple performance features. Using the tool in a study of eighth notes inégalité, a relationship between timing and loudness was found.

Keywords

Synthetic Performance, Notes Inégales, Timing, Articulation, Duration, Loudness, Dynamics

1. INTRODUCTION

Up to now, timing seemed to be a core feature of expression in music. Despite large phrase arches and ritardandi, it is the temporal shape of small figures, which is crucial for the impression of liveliness and expressiveness. Gabrielsson analyzed ratios of beats or frequent notes at the sub-beat level [6]. He discovered that notes of equal value are performed with different ratios—even in several rhythms of a broad stylistic range, reaching from Swedish folk songs to the Viennese Waltz. Similar phenomena were also discovered in more remote cultures, as Gerischer demonstrated [7], or in diverse Western music styles, which were analysed, for instance, by Langner [11].

The knowledge of an unequal shaping of notes has been known in theory and practise for centuries. A prominent instance is the playing of "notes inégales" in French Baroque, which is discussed in detail by Hefling [8]. Simply put, playing eighths notes inégal means that a note on the beat lasts a little longer than its actual value. The time added on this note is subtracted from the succeeding eighth note between the beat. Inégalité therefore describes a particular phenomenon of timing. However, referring to original sources like Quantz' treatise [13], inégalité has always included aspects of loudness and articulation. Inégalité therefore could be seen as an emphasis of a note, which is prominently achieved by lengthening the first eighth note (timing), but also by playing it louder (loudness) or articulating it differently, e.g. by manipulating the duration of a note. An analysis of inégalité should therefore take into account the complex interplay of at least timing, loudness, and tone duration.

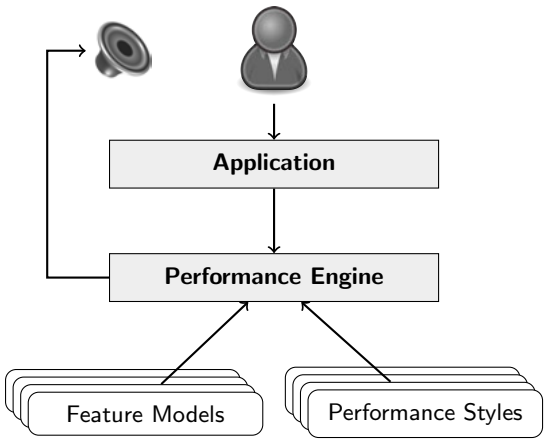


Figure 1: The technical setup of the study.

To focus on interdependencies between these three performance features, we developed a tool, with which listeners could adjust these parameters interactively. (see Section 2 and 3). The main idea is to extend the common "Analysis-by-Synthesis" approach that was outlined in detail by Bengtsson and Gabrielsson [1] and reaches back to Seashore and his colleagues, like Metfessel [14, 12]. Normally, listeners are asked to judge several stimuli, which comprised synthetic performances that differed in some particular performance parameters. The listener's judgements then indicate which parameters are most appropriate.

This approach had to be modified, for the total amount of stimuli depends on the number of grades a parameters is subdivided into and increases to the power of parameters used. This would mean that a combination of three parameters of 21 grades each would result in $21^3 = 9261$ stimuli, an amount impossible to be judged by listeners. The task was to provide all stimuli but at the same time reduce them to a minimum for each participant. This was done by letting the participants manipulate these three parameters independently and interactively until they reached the parameter combination they preferred.

The following Section includes a description of the tool. Section 3 describes the method of the inégalité study. Finally, a general discussion closes the paper in Section 5.

Figure 2: First bars of the Polonoise taken from the overture in d-major "La Gaillarde" TWV 55:D13 for strings & b.c., composed by G. P. Telemann.

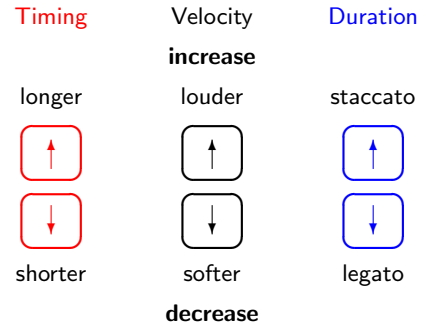


Figure 3: Instructions for the modification of eighth notes occurring on the beat.

2. TECHNICAL SETUP

We have developed a performance rendering engine that allows to create expressive music performances based on a high-level description language (XML-based). Different performances can be described and are stored as *performance styles*. These are rendered into expressive MIDI sequences. The performance engine furthermore allows to interactively change over to another performance style while the music plays. It automatically creates musically plausible seamless transitions by simulating the reactions of musicians with an agent-based approach [2]. This allows any application to interactively control the music performance just like in the case of the study described in this paper. The participants had to explore a domain of possibilities and choose the location that matches their preferences. The setup is shown in Figure 1.

Formally, expressive performance can be regarded as a series of transformations that is applied to a musical composition. We distinguish three categories:

Timing defines the mapping from symbolic time, which is where the composition is defined in, onto physical time (usually in milliseconds). It combines several layers of timing features, i.e., tempo (macro timing), rubato (self-compensating micro deviations), asynchrony (between parts in a polyphonic setting), and random imprecision. [3, 4]

Dynamics sets the loudness of each musical event. It incorporates a macro layer that defines the basic loudness function of the musical piece, and two micro layers that introduce fine deviations to the basic loudness. Micro dynamics features are metrical accentuations and articulations (only the loudness aspect of the latter). [5]

Articulation deals with the way each musical event (note) is played. Articulation instructions affect the loudness, duration and timbre of the notes to be articulated. [9, 10]

We have developed mathematical models to describe, analyze, and synthesize these features. Their flexible parameterization allows for a huge bandwidth of characteristics, including those that could be observed with human musicians, and even atypical ones that are more of theoretical interest. Most models do even allow to define new instances, e.g., new articulations, accentuation schemes, and rubato patterns. This flexibility and full control over any performance detail, even the most subtle, makes our performance engine a very useful tool especially for musicological and psychological experiments and studies.

3. METHODOLOGY

Timing, loudness, and duration were tested by the use of a four bar phrase taken from a Polonoise by G. P. Telemann (see Figure 2).

Two tests were carried out, first a *separate parameter test* and then a *combined parameter test*. In the first test the parameters were analysed separately. The participants were asked to modify the performance of eighth notes labelled in a score (as shown in Figure 2) of the stimulus, which was presented in a loop. The eighth notes on the beat (□ in Figure 2) were emphasized by pressing an Arrow-Up key. A decrease of this emphasis or even an emphasis of the eighths between the beats (▲ in Figure 2) was set by pressing the Arrow-Down key. At the limit of the parameter spectrum a beep signalled that no further modification was possible in that direction.

The first test consisted of two tasks: First, the participants were asked to identify the parameter they set. Then they had to tune the relation of the eighths as they supposed to sound best and confirm with the Enter-key. All participants only saw the score including the squares and triangles as shown in Figure 2 and therefore depended completely on an auditive feedback.

In the second test the same stimulus was presented. The participants now had the opportunity to tune all three parameters independently at the same time. As Figure 3 shows, all parameters were set by an array of six control-keys above the arrow keys, which were labelled with up and down arrows in red, black and blue.

3.1 Feature Manipulation

Our performance engine implements two output modes, a standard MIDI mode and further functionalities for the output over the software sampler *Vienna Instruments* that utilizes some its advanced possibilities in controlling certain tone parameters. In this study, however, we applied the software *VSampler 3* that runs smoothly on laptops, allowing for more flexibility and mobility.

Via the study application the participants of the study had control over certain parameters that influence the timing (rubato), dynamics (metrical accentuation), and articulation (tone duration). The interface is shown schematically in figure 3. The parameter space was discretized into 21 steps (controller states) reaching from -10 (minimal setting) to 10 (maximal setting).

Rubato: Rubato is a timing distortion that is compensated within a certain timeframe. Hence, the basic tempo remains unchanged at 120 bpm. In this study the rubato frame was of the length of a quarter note and repeatedly applied over the whole musical piece. For each quarter note a swing-like distortion could be cre-

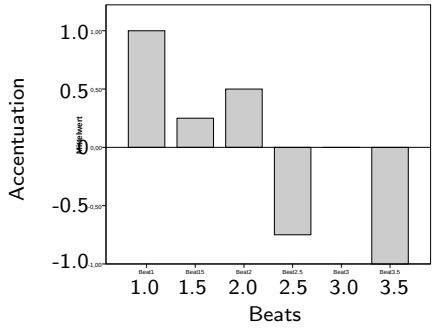


Figure 4: The accentuation scheme could interactively be scaled.

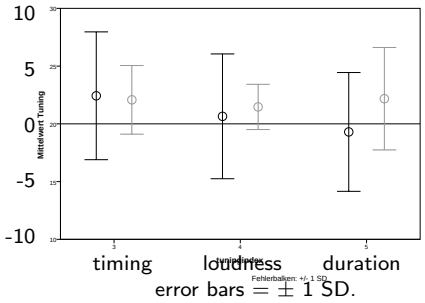


Figure 5: Results of separate tuning. Values tuned by participants identifying parameters correctly (gray bars) compared to others (=black bars).

ated. This distortion is modelled by a potential function in the unity square. The timing relation between first and second eighth is:

$$\left((0.5^i)\,0.98 : (1-0.5^i)\,0.98\right) \quad | \quad i \in [0.4, 1.6]$$

The timing controls enabled the probands to set the parameter i. Values between $10 \rightarrow (i = 0.4)$ and $-10 \rightarrow (i = 1.6)$ were linearly interpolated.

Metrical Accentuation: On the level of micro-dynamics an accentuation scheme was defined (see figure 4) and applied to each measure. The participants were able to set its intensity, that is the loudness scale of the scheme, ranging -60 up to 60 MIDI velocity units. The possible settings reached from 10 to -10, whereas 10 created a pianissimo for the softest and a fortissimo for the strongest accentuations (a range of 107 MIDI velocity units) and -10 caused a likewise pronounced but inverted accentuation scheme.

Articulation: The probands could set the durations of the notes from legato to a very short staccato. This cause either a very cantabile or a rhythmically pronounced performance. However, the behaviour of articulation settings was even more complex. The duration of each second eighth note under a quarter beat decreased faster than the first so that the emphatic relation between both shifts accordingly. Three sampling points were defined (see the following table) and linearly interpolated.

controller state	Duration of 1st eighth	Duration of 2nd eighth
10	0.35	0.2
0	0.7	0.4
-10	1.0	1.0

In the *separate parameter test* the controllers were randomly initialized at the extremes, i.e. -10 or 10, whereas, in the *combined parameter test* the initialization was random over the full parameter space. Positive controller settings created plausible features that could also be observed within human performances. With regard to metrical accentuation and rubato, the negative settings created implausible performances.

All interactions were logged, as well as the test duration and the initial and final controller settings that the participants approved. This allows insights into how the users explored the search space and how long they listened to certain settings until they made a decision and interaction.

3.2 Participants

The participants comprised 36 western socialized adults, including 21 women and 15 men, 10 professional musicians specialized in Baroque performance, 16 with a degree in music, musicology, music pedagogy, church music or similar, and 20 playing an instrument for more than ten years.

4. RESULTS

The answers given in the identification task were collected and manually classified as correct, ambivalent, or incorrect. A correct answer had to be unambiguous. This was a problem regarding the differentiation between timing and duration, for the term "length" is ambivalent.

Unfortunately, the distribution of controller values from the second task was not normal. A possible reason was that obviously some participants did not use a spectrum large enough to get an impression of the possibilities they had (it was also hardly possible to identify the parameter if it had not been modified to a certain degree). Therefore, data were excluded when the modification range during the test was below seven (regardless of the final value). This was the case for two samples—the remaining samples showed a normal distribution.

No influence of expertise on the parameter identification or the adjustment was found, which might be caused by the small amount of professional musicians. However, the label "expert" was restricted to the individual musical background. But those identifying the parameters correctly or ambivalent differed to the remaining in the parameter adjustment: the variances of timing and loudness significantly decreased. As it turned out, the standard deviation sd for participants identifying loudness correctly or ambivalent was $sd = 1.69$ compared to $sd = 5.4$ for those who did not identify what they had modified (with a significance of $p = 0.001$ in a Levine test). For timing these differences were smaller ($sd = 2.95$ in the correct group and $sd = 5.53$ in the incorrect group) but still significant (with $p = 0.026$ in a Levine test). The differences are plotted in Figure 5.

Generally, the difficulties the participants had were more pronounced than expected. Hence, 11 participants did not take part in the second test. The remaining 25 participants included all professional musicians. 19 participants of the second test played an instrument for more than ten years and 15 had a degree in a music related subject.

A correlation analysis uncovered a highly significant negative correlation between timing and loudness (with $r = -0.653$ and $p = 0.001$). The plot in Figure 6 shows every sample pair of timing and loudness values, and a regression line to illustrate the correlation. In contrast, duration was correlated neither with timing nor with loudness. Neither

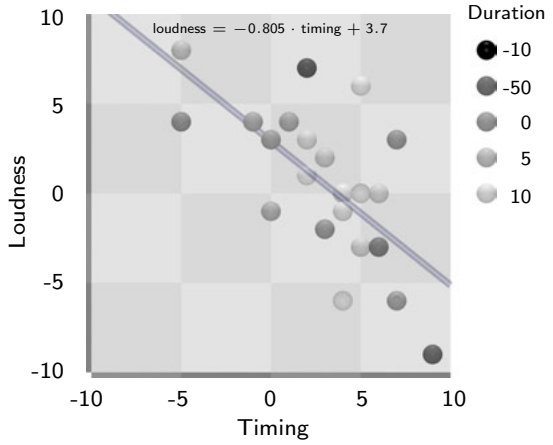


Figure 6: The regression line demonstrates the negative correlation between timing and loudness values.

a t-test nor a Levine-test could detect a difference between the separate and combined modulation of any parameter.

5. GENERAL DISCUSSION

The tool described in the previous Sections allows listeners to find a preferred stimulus out of a large number of stimuli. Generally, the tool can be used for diverse designs of studies with an Analysis-by-Synthesis character. The advantage is that multiple performance features as well as their interdependencies can be tested. In the present study the participants easily understood the tasks and handling of the study application, which is important for listeners unacquainted with alternative interfaces and hardware. Different research questions can nevertheless require different interfaces, which is also not a problem, for the user interface and the performance engine are separate modules. Participants could use sliders instead, move around in a room or produce physical gestures, which can trigger the application. The latter in particular offers interesting perspectives for therapy and rehabilitation, since the setup of stimuli can prompt participants to move in a particular way and avoid other movements. The feature models and performance styles are freely editable and exchangeable. This is particularly interesting because a large amount of stimuli allows a feeling of continuous stimulus modification. This also induces an impression of the direction of change, which the user can consciously track or reverse.

However, the study described focused on interdependencies of timing, loudness, and duration with respect to "inégalité" in eighths notes performances. The results can lead to the suggestion that multiple means for emphasis can be cumulative: an intensive use of a single performance parameter causes the same emphasis—and therefore the same impression of inégalité—as a slight use of multiple performance parameters. To avoid an overemphasis, the musician or music producer must balance out these parameters, which leads to a compensation effect.

The difficulties occurred in the study of "inégalité" uncovered a problem more relevant to a rather musicological discussion. Many participants could not decompose the parameters that are manipulated to achieve inégalité, even if they recognized that something was changing and intended as well as achieved an inégalité while exploring the range of stimuli.

Acknowledgments

We thank all our participants from the department of early music of the UdK in Berlin, the "Zentrum für Telemann-Pflege und -Forschung" in Magdeburg, the Magdeburgian Music school, the Orchestra of the "Magdeburger Musikfreunde", the "Otto-von-Guericke-Universität Magdeburg" as well as those who do not belong to any of the above mentioned institutions.

6. REFERENCES

[1] I. Bengtsson and A. Gabrielsson. Rhythm research in Uppsala. *Music Room and Acoustics*, pages 19–56, 1977.

[2] A. Berndt. Decentralizing Music, Its Performance, and Processing. In M. Schedel and D. Weymouth, editors, *Proc. of the Int. Computer Music Conf. (ICMC)*, pages 381–388, New York, USA, June 2010. International Computer Music Association, Stony Brook University.

[3] A. Berndt. Musical Timing Curves. In *Proc. of the Int. Computer Music Conf. (ICMC)*, Huddersfield, UK, Aug. 2011. International Computer Music Association, University of Huddersfield. in review.

[4] A. Berndt and T. Hähnel. Expressive Musical Timing. In *Audio Mostly 2009: 4th Conf. on Interaction with Sound—Sound and Emotion*, pages 9–16, Glasgow, Scotland, Sept. 2009. Glasgow Caledonian University, Interactive Institute/Sonic Studio Piteå.

[5] A. Berndt and T. Hähnel. Modelling Musical Dynamics. In *Audio Mostly 2010: 5th Conf. on Interaction with Sound—Sound and Design*, pages 134–141, Piteå, Sweden, Sept. 2010. Interactive Institute/Sonic Studio Piteå, ACM.

[6] A. Gabrielsson. The Performance of Music. In D. Deutsch, editor, *The Psychology of Music*, pages 501–602. Academic Press/Elsevier, San Diego, 1999.

[7] C. Gerischer. *O singue baiano - Mikrorhythmische Phänomene in baianischer Perkussion*. Peter Lang, Frankfurt a.M., 2003.

[8] S. E. Hefling. *Rhythmic Alteration in Seventeenth- and Eighteenth-Century Music, Notes Inégales and Overdotting*. Schirmer Books, New York, 1993.

[9] T. Hähnel. From Mozart to MIDI: A Rule System for Expressive Articulation. In *Proceedings of New Interfaces for Musical Expression (NIME2010)*, pages 72–75, Sydney, Australia, June 2010. University of Technology Sydney.

[10] T. Hähnel and A. Berndt. Expressive Articulation for Synthetic Music Performances. In *Proceedings of New Interfaces for Musical Expression (NIME2010)*, pages 277–282, Sydney, Australia, June 2010. University of Technology Sydney.

[11] J. Langner. *Musikalische Rhythmen und Oszillation – Eine theoretische und empirische Erkundung*. Schriften zur Musikpsychologie und Musikästhetik - Band 13. Peter Lang, Frankfurt a.M., 2002.

[12] M. Metfessel. Sonance as a form of tonal fusion. *Psychological Review*, 33(6):459–466, 1926.

[13] J. J. Quantz. *Versuch einer Anweisung die Flöte traversière zu spielen*. Bärenreiter, Berlin, 1752. Faksimile-reprint (1997).

[14] H. Seashore. The Hearing of the Pitch and Intensity in Vibrato. In C. E. Seashore, editor, *The Vibrato*, volume I of *Studies in the Psychology of Music*, pages 213–235. University Press, Iowa, 1932.

1city1001vibrations : development of a interactive sound installation with robotic instrument performance

Sinan Bökesoy
composer &multimedia artist
Büyükada, Istanbul
sinan@sonic-disorder.com

Patrick Adler
Robot Engineer
Augsburg, Germany
paadler@gmx.net

ABSTRACT

"*1city1001vibrations*" is a sound installation project of Sinan Bökesoy. It does continuous analysis of live sounds with the microphones installed on top of significant places at Bosphorus - Istanbul. The transmitted sounds are accompanied by an algorithmic composition derived from this content analysis for controlling two Kuka industrial robot arms performing the percussions installed around them while creating a metaphor through an intelligent composition/performance system. This paper aims to focus on the programming strategies taken for developing a musical instrument out of an industrial robot.

Figure 1. The installation venue in Istanbul, July2010

Keywords

Sound installation, robotic music, interactive systems

1. INTRODUCTION

Robots capture great interest while creating an existence by providing physical and visual cues to the audience. (Figure 1) Our aim was creating a metaphor by translating the Bosphorus sounds to the vibrations of percussion surface of various ethnic percussions. As example applications using robots, there are mechanical systems offering one degrees of freedom by applying the hit stroke to a fixed point of percussion surface within the dynamics of a hammer action [3][6][7]. We used an industrial robot with 6 degrees of freedom offering the movement potential close to the human arm. A communication protocol with semantic description of the robot control commands has been developed with an interface between the robot software and the controlling musical application.

1.1 Listening to the "Bosphorus" sounds

The microphones, as the 'ears' of the robots do transmit the signals as internet audio stream via an AudioIP unit. This audio-stream is being interpreted by an analysis application (Figure 2) with the computer at the installation venue.

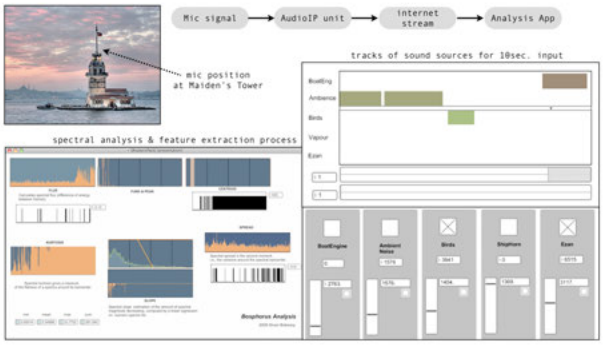


Figure 2. Transmitting the Bosporus sounds to the analysis application and tracking of the sound sources on Max/MSP

The analysis software is a Max/MSP[2] application and based on the interpretation of the spectral analysis of the audio stream with the help of the powerful Zsa.Descriptors [5]. The feature extraction process from the spectral data aims to recognize 5 specific sound sources found on everyday routine of the Bosphorus. The analysis software is reporting the onset times of the sound events we are interested in. Each tracking of the relevant sound source is used to create a percussion score according to the rules defined for the translation of Bosphorus sonic space to percussion gestures.

The input of the interactive sound system is the content analysis of the audio-stream coming directly from Bosphorus. The output of this patch is the percussion score to be performed by the robots and also some of signal processing tools to create a dispositive sound distribution at the installation venue. A 10sec. long audio analysis frame will be captured live and the analysis will be sent after each 10sec. frame to the algorithmic composition patch. Therefore the performed percussion score belongs to 10sec. in the past. Each robot has access to 2 types of Istanbul cymbals, 1 gong, one glockenspiel, 1 davul (a ethnic drum), 1 bendir (a ethnic hand drum) and 2 tibetian bowls. The generated rhythmic pattern for the relevant instrument is influenced by the length of the tracked source, and also on some statistical distribution of spectral parameters. The translation of percussion score is achieved by dedicated XML strings, which will be interpreted by the robot software. Each of the five tracks corresponds to a sound source in the analysis.

The application selects the tracks belonging to the sound sources occurring with highest density in each 10sec. There are 39 different gestures programmed on the robot software, and these can be called with appropriate parameters to modify the gesture according to the performance needs.

2. THE DEVELOPMENT PHASE

We use 2 Kuka KR16 industrial robots having 6axis motion capability, 1.5m reach capacity and up to 15kg load capacity. First, we intended to investigate the dynamics of drumming with sticks, since the percussion timbre is highly dependent on the factors including contact area, hit damping duration and pressure on the drum skin. By being able to control these parameters and simulate them with robot gestures we can achieve the timbre richness and sonic variety for each instrument.

2.1 The robot gripper part

The gripper of the robot is attached at the flange of the robot, positioned next to the 6th axis motor A6. The drum sticks consist of parts named as the *butt, shaft*, *shoulder* and the *tip*. In our gripper design; the butt part, which would be grasped normally by a human hand, is positioned inside a spring for holding the stick tightly. (Figure 3) Our gripper can hold 3 different sticks; wooden, soft and plastic stick. The robot can turn the gripper with the A6 motor, so that the chosen stick can hit on the vertical axis to the percussion surface. On Figure 8, there is the list of the programmed gestures along with the selected stick type for each instrument and the impact points.

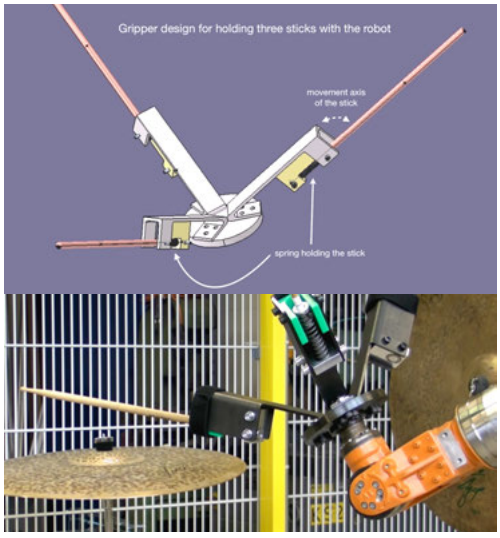


Figure 3. The gripper design (by GD Engineering, Germany) and its application

The mechanism of the "hit gesture" is inspired by a piano hammer action and it is quite simple for quick adjustments. For each stick we have chosen an appropriate spring to realize the required elastic hit motion applied by the gripper. (Figure 3)

2.2 The robot programming part

Kuka Kr16 has 6 degrees of freedom, and the movement on each axis is driven by an independent motor labeled from *A1* to *A6*. To control the motion of the robot each given value is compared to the actual angle of each axis and regulated by motion controllers for each axis. Among the two basic motion types; [4]

PTP (point to point) movement: Each axis uses the shortest way from the actual position to the destination position. PTP can be programmed directly with values for each axis (eg "*A1 30*" in degrees) or with cartesian values referring to a given base (eg "*X 100*" in mm). (Figure 4)

LIN (linear) movement: each axis is controlled for a joined movement; they start and stop its movement at the same time and the slowest axis will cause the others to slow down. The movement is a direct line from the actual position to the destination point.

The cartesian movements have to be related to a base origin, which are referred to the "robot root". The gripper hot point (the sticks end) has to be defined and are always referring to one base except from the *PTP* movements with the axis values in degrees. The rotation, the speed and acceleration for each movement has to be defined within the motion command. KUKA offers spline motion path since software version 5.5, but with the software version we had, our robot-percussion gesture ended up with linear movement path.

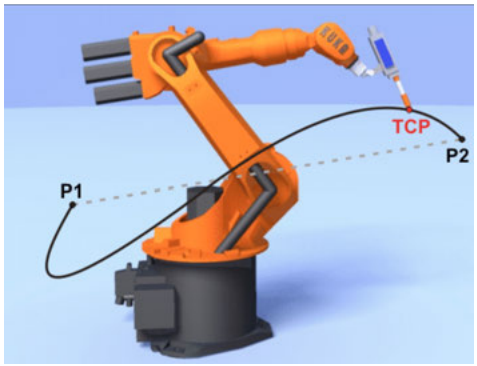


Figure 4. PTP movement: Robot axes are rotational; curved paths can be executed faster than straight paths.

2.2.1 Communication interface

We have chosen the *KUKA Ethernet KRL XML* platform to develop our communication interface. Both the MaxMSP and the KUKA Ethernet Interface are working as TCP clients, and it was necessary to develop a bridge application, which receives the commands from MaxMSP and sends them to the robot. The maximum speed of ToolCentralPoint transfer is 2meters/sec. After establishing the basic command transfer, we defined the parameter space for each command according to the dynamics of the "hit gesture" and the common XML structure. (Figure 5)

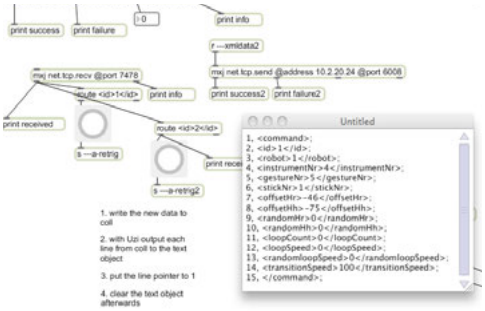


Figure 5. The Max patch building the XML messages and sending to the robot via TCP/IP protocol.

2.2.2 *The dynamics of the "hit gesture"*

The programming of the robot was divided into two parts. First the received commands had to be read, stored to variables and then verified. The stored values were used to recognize the selected instrument and to decide which movement had to be taken in order to move to the instrument without collisions. Then, the stroke parameters had to be applied to a common stroke movement routine. According to the robot coordinate system the impact point of the stick is the end point of the arm. The robot utilizes the 3 transition points to move the arm between instrument groups: for the cymbals *Pt1*, for the drums *Pt2*, and for the bowls and the glockenspiel *Pt3*. From each transition point, it moves to the relevant *Hr* points for the instrument it will hit at the *Hh* point (Figure 6). The *Hr* point is the state of the robot, where it waits ready to apply the "hit gesture" *Hr - Hh - Hr*.

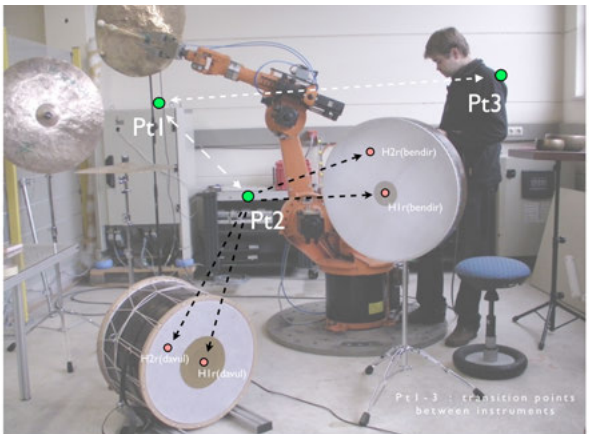


Figure 6. Robot engineer P. Adler testing the software. The robot moves itself between various taught points.

The kinetic hitting force caused by this movement is stored on the spring holding the relevant stick, and then the robot stops necessarily at the *Hh* point but the spring releases the stored energy and the stick continues and hits the surface. This is a basic spring oscillation mechanism applied on the stick axis. (Figure 7)

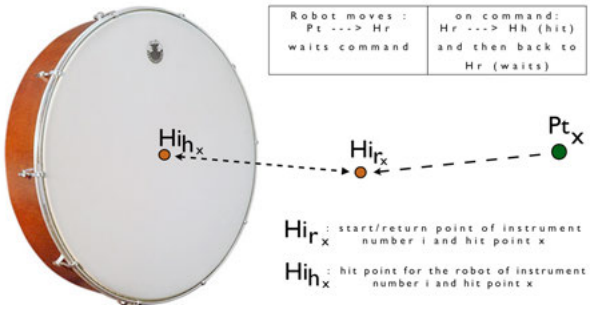


Figure 7. According a command, the robot passes through a transition point and then a *Hr* point to hit at *Hh* point

We proceeded by characterizing the percussion gestures, which will be applied by the robot arm. (Figure 8) Each instrument got its own base (coordinate system) and a common base for the movements between the instruments. Each gesture or transition movement was referred to these bases.

We have developed a special subroutine for the gestures, which uses the quicker/faster axis motors on the arm (A4 - A6, especially A5) for the basic stroke movement and added motion in the kinematics through (A1 - A3) to increase the stroke hardness. This strategy does use the advantages of a robot arm with 6 degrees of freedom against the mechanical arms with one axis movement capability, and the richness of the movement does translate into sound quality. [1]

Figure 8. A catalogue of percussion gestures

Our expectations from the robot drummer were reformed with meaningful optimizations of our model according to the dynamic motion and kinematic boundaries of the robot. The deceleration of the robot was in this case (small and therefore fast robot) adequate. Thanks to the spring mechanism there was no negative effect on the speed of the stick.

2.2.3 *The parameter space of the "hit gesture"*

To achieve the natural dynamics of percussion performance, the robot should be able to adjust the sound quality of its gestures in several manners. The analogous method is by striking the percussion surface at different locations with different sticks and contact points. The loudness variety is achieved by hitting harder or softer, which is the amount of pressure on the surface. The angle of the stick on the impact point is also relevant. We have implemented some controllable parameters, which can adjust the position and behavior of the robot arm to modify the impact quality. The following parameters can be adjusted directly within our musical application.

- *Stick type* : 1- Woodenstick 2- Softstick 3- PlasticStick
- *Destination address* : *(Pt) (Hr)* All transition points and hit gesture start/return points available for each instrument are the possible positions of the robot. Instrument number *i* is specified like 1 for Davul 2- Bendir 3- Gong etc.. Hit positions for each instrument are sub categorized as *P1, P2, P3* etc. (Figure 8)
- *Speed* : is a percentage of the maximum speed of the motors driving the axis movement specified between *0* and *100*. The robot specs give the speed in terms of angular speed.
- *Hr position off*: offset value to move the *Hr* position towards the percussion surface or away. 0 is the original taught point.
- *Hh position off*: offset value to move the *Hr* position towards the percussion surface or away. 0 is the original taught point.

Repositioning the *Hr/Hh* points are of crucial importance in order to modify the sound quality achieved. For instance, when both the *Hr* and *Hh* points are modified with the same amount of offset value, the result is hardening or softening the impact while maintaining the speed of the robot movement.

[1] www.c-av-e.com/Kukarobot-bendir.mov

- *Gesture Type* : 0 – Basic Shot, 1 – Full Loop ; With "basic shot", the robot hits the surface and goes back to the Hr point and maintains its position until a next command. When "full loop" is selected, the robot repeats the hit sequence with the specified amount.
- *Loop amount* : The robots does repeat the hit sequence this number of times and the *Hr*, *Hh* and *speed* parameters can be adjusted inside this loop to achieve performance techniques such as accelerando, crescendo with percussion tremolos.

We have also implemented on the robot software controlled randomness for the *Hh* point in terms of adjusting its coordinate on the horizontal plane of the percussion surface, which would humanize the impact since otherwise a robot would always hit to a precise point. During the evaluation process our first hand guide is always our ears, hence it was easy to observe whether the representation of the robot performer model was sufficient to finely represent the subtility of required gesture dynamics or not. We have created a database for each gesture for the easy access of the gesture types with specific dynamics; soft hit, hard hit, slow tremolo – soft hit, fast tremolo fast hit etc.

```
1, 1 1 1 -48 8 33 46 0 -12 4 9 0 -12 0 -12 -75 -33 -26 -7 -75 -33 -26 -7 0 0 570 640 7704 3250 3250 3250;
2, 1 2 1 -56 -43 5 1 -26 -11 -13 6 -13 6 8 2 -16 -37 -16 -1 -59 -77 -53 -20 570 570 5160 9970 4860 9568 2150 3974 2150 4028 1754 3200 1686 3192 1716 3222 1810 3352;
3, 1 1 2 -74 -15 -5 3 6 16 9 22 9 13 13 32 -24 -32 13 32 -24 32 13 32 620 700 6116 12000 6200 12200 2640 5050 2820 5450 2100 3950 1830 4200;
4, 1 2 2 -93 42 32 59 1350 1288 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
5, 2 1 1 -56 -5 -85 -12 20 2 1 9 7 0 37 -2 2 -13 9 8 -40 -38 10 -7 530 600 8100 16000 7900 15600 3000 5910 3300 6350 2200 4200 2200 4270 1940 3754 1800 3300;
6, 2 2 1 -30 -25 -27 31 2 13 -3 32 -3 -19 -3 39 -3 16 13 23 19 1 -8 30 450 500 4600 8850 4500 8682 1940 3722 2230 4230 1560 2860 1700 3200 1644 3040 1620 3000;
7, 2 3 1 -78 -5 -32 16 1230 1160;
```

2.3 Event scheduling

In a musical performance knowing precisely each process time is necessary. When the robot receives a command, it sends back an acknowledgement to the MaxMSP application. (Figure 9) It does also send task finished messaged when the robot arm returns back to the specific *Hr* position. We define the time of command receive as $T_{command(x-1)}$ for the hit event (x-1). Then the total task period, which lasts for the "hit gesture" is;

$$T_{return(x-1)} - T_{command(x-1)} = t_{task(x-1)}$$

We do calculate this time interval easily by starting a clock with the send of the command and stopping it with the task end message received from the robot. This is a full cycle as shown on Figure 8. We need to know, the time interval between the command receive and the surface contact of the robot precisely. We name the moment of impact as $T_{hit(x-1)}$, and the hit process interval can be calculated with;

$$T_{hit(x-1)} - T_{command(x-1)} = t_{hitproc(x-1)}$$

We use a microphone to capture precisely the act of surface impact on the percussion. For instance with the help of the maxmsp bonk~ object[8] we can detect the hit event to stop the clock. Between $T_{command(x)}$ and $T_{return(x-1)}$ the robot will be busy. The musical application has to consider this fact. Therefore we had to catalogue all possible $t_{hitproc(x)}$ and $t_{task(x)}$ values, which means $C(37,2) = \frac{39!}{37!2!} = 741$ values to be stored for both. For this purpose, another max patch has been used[2] in to automate this process. The patch measures all 741 $t_{hitproc(x)}$ and $t_{task(x)}$ values and prepares a matrix.

Therefore we define much time in advance we need to send the command to the robot at its last position to get a hit on a certain time $t_{(x)}$ and what would be the minimum task duration.

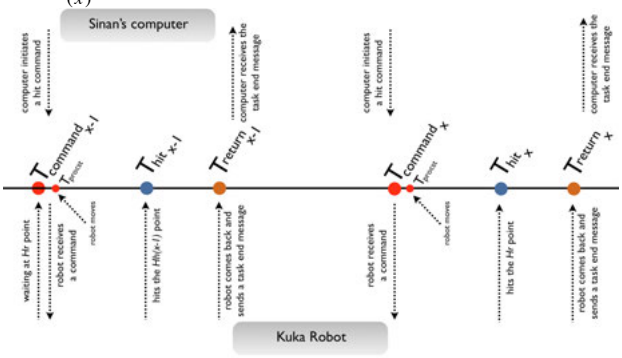


Figure 9. The chronological view to the event schedule and message exchange between 2 computers.

A high level musical representation of this structure has been required for direct score –to- robot gesture translation. We have chosen Ableton Live[1] with the Max4L addon, since it can directly integrate a MaxMSP patch, which does translate the incoming sequencer messages to XML messages interpreted by the robot software. The track midi information is translated to relevant XML strings with proper time stamping of events as explained on section 2.3 by checking the $t_{hitproc(x)}$. A midi note represents a percussion gesture. Continuous midi controller messages assigned to it are translated to real-time parameters such as gesture speed, *Hr* and *Hh* position of the stick and the loop amount.

3. ACKNOWLEDGEMENTS

My thanks to Kuka Robotics, Germany, which has supported the project by providing us 2 KR16 robots and also has assisted to the organization of robot motion programming and logistics. This project has been realized with the financial support of *Istanbul2010 European Capital of Culture Agency*.

4. REFERENCES

[1] Ableton – Ableton Live Suite, www.ableton.com
[2] Cycling74 – MaxMSP, http:www.cycling74.com
[3] Kapur A., Darling M. A Pedagogical Paradigm for Musical Robotics. In *Proceedings of the Conference on New Interfaces for Musical Expression*, (Sydney, Australia, 2010)
[4] Kuka Robot Group, *Kuka System 5.4 Software Reference Manual.* Issued on 24.04.2008.
[5] Malt, M. and Jourdan, E. Real-Time Uses of Low Level Sound Descriptors as Event Detection functions Using Max/MSP Zsa.Descriptors. In *Proceedings of the SMCM9 conference.* (Recife, Brasil, 2009)
[6] Singer, E., Feddersen, J., Redmon, C. and Bowen B. LEMUR's Musical Robots. In *Proceedings of the NIME Conference* (Hamamatsu, Japan, 2004)
[7] Weinberg, G., Driscoll S., Parry M. Musical Interactions with a Perceptual Robotic Percussionist. *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*. (Nashville, TN 2005).

[2] www.c-av-e.com/gest-walk.mov

The medium is the message: Composing instruments and performing mappings

Tim Murray-Browne
Queen Mary University of London
Centre for Digital Music
School of Electronic Engineering and Computer Science
tim.murraybrowne@eecs.qmul.ac.uk

Di Mainstone
Queen Mary University of London
qMedia Artist in Residence
School of Electronic Engineering and Computer Science
dimainstone@hotmail.com

Nick Bryan-Kinns
Queen Mary University of London
Interaction, Media and Communication Group
School of Electronic Engineering and Computer Science
nick.bryan-kinns@eecs.qmul.ac.uk

Mark D. Plumbley
Queen Mary University of London
Centre for Digital Music
School of Electronic Engineering and Computer Science
mark.plumbley@eecs.qmul.ac.uk

ABSTRACT

Many performers of novel musical instruments find it difficult to engage audiences beyond those in the field. Previous research points to a failure to balance complexity with usability, and a loss of transparency due to the detachment of the controller and sound generator. The issue is often exacerbated by an audience's lack of prior exposure to the instrument and its workings.

However, we argue that there is a conflict underlying many novel musical instruments in that they are intended to be both a tool for creative expression and a creative work of art in themselves, resulting in incompatible requirements. By considering the instrument, the composition and the performance together as a whole with careful consideration of the rate of learning demanded of the audience, we propose that a lack of transparency can become an asset rather than a hindrance. Our approach calls for not only controller and sound generator to be designed in sympathy with each other, but composition, performance and physical form too.

Identifying three design principles, we illustrate this approach with the Serendiptichord, a wearable instrument for dancers created by the authors.

Keywords

Performance, composed instrument, transparency, constraint.

1. INTRODUCTION

The possibilities of new computer-based musical instruments are vast. As well as the unlimited sound possibilities of software instruments, there is hope they may be made easier to learn [18] and even advance to role of co-performer [2]. However, many fail to enrol performers beyond their creators or engage audiences beyond those within the field.

Research investigating this problem often considers the experience of a potential performer. It highlights capabilities of traditional instruments that may be neglected, such as reliably reproducing an output, a diversity of musical outputs and a fine degree of control allowing performance nuances [10]. At the same time, it is hoped new instruments might be easier to learn than traditional ones, providing a quicker and steadier path of advancement [10, 18]. Some create new instruments more as an extension of their compositional practice, arguing that rather than building 'super instruments,' projects should be driven by specific compositions [3]. In this sense, composing may be seen as defining a musical space to be explored in performance [12].

Others focus on the experience of the audience. Fels et al. [6] consider the difficulty of establishing *intimacy* between audience and instrument without the commonly understood modes of performance possessed by traditional instruments. Lack of established performance practice can make it difficult to interpret (or even define) stylistic variation [7].

In this paper, we explore the difficulties that novel instruments face from the audience's point of view. By analysing performance as an act of communication we arrive at an approach to instrument creation and performance that draws upon their novelty as a strength rather than a hindrance.

2. ISSUES FACED WHEN CREATING NOVEL INSTRUMENTS

2.1 Transparency

Transparency is defined as 'the psychophysiological distance, in the minds of the player and the audience, between the input and output of a device mapping' [6, p. 109]. For example, whilst someone watching a violinist perform may not understand how each note is fingered, the connection between action and sound is intuitive and familiar through previous exposure to the violin; hence it is considered transparent. New instruments with unconventional mappings that cannot rely upon this prior knowledge are initially opaque to the audience. The challenge is then to con-

NIME'11, 30 May–1 June 2011, Oslo, Norway.

struct instruments whose modes of use seem 'inevitable' [11] through methods such as drawing upon embodied metaphors – familiar day-to-day gestures and interactions [5].

But whilst that may *facilitate* transparency, an audience new to the instrument will still have the extra mental burden of learning how it works as well as appreciating the music it produces. Consequently, the musical content of a performance may be simplified to prevent it becoming too difficult for people to appreciate [15]. However, following this path too far risks reducing the music of a performance to a tool to facilitate the demonstration of the instrument. It reduces the appeal of returning to see another performance as, chances are, the instrument will be exactly the same and the music still simple for the sake of newcomers.

2.2 Engaging the audience

Many theories of musical appreciation focus on *expectation* (e.g. [14]). Someone listening to music is constantly and subconsciously creating a model to anticipate how it will progress. Their enjoyment depends upon both successful prediction affirming their model and a degree of surprise allowing it to develop and improve [9]. Music, in this context, becomes an act of communication, balancing the novel with the familiar. This balance is analogous to Csikszentmihalyi's theory of *flow* [4], the 'optimal experience' that can be attained when performing at the limits of one's ability. When the difficulty of a task greatly exceeds ability, anxiety follows; when ability greatly exceeds difficulty, boredom follows. But when difficulty matches ability, the person is both understanding and being challenged, allowing them to be engaged fully, improve their ability and achieve flow.

Witnessing a musician *in flow* during a performance is of course important. But the consequences of flow are also relevant when considering audience engagement. Sherry [16] identifies the difficulty of media as the degree to which it deviates from the formal convention of a familiar genre. Too much deviation can seem chaotic. It is difficult to identify any structure and frustration arises. Too little deviation and the media seems trite. Its structure has no surprises making it predictable and boring. In both cases it is a lack of *emerging structure* – improvement in our ability to predict – that hinders audience engagement. When there is just the right amount of deviation a structure steadily emerges that both affirms and expands upon our prior experience.

We may consider the workings of traditional instruments – how they are played and relate gesture to sound – as norms of musical performance. Thus, performing a novel instrument is a departure from convention, challenging the audience to understand it, to develop transparency and observe an emerging structure. The instrument is no longer just a tool to create music but is itself a part of the show.

2.3 The double bind

And herein lies the double bind of novel instruments: they seek to be both a tool to perform music and part of the musical composition itself.

The instrument as a tool

The instrument as a tool seeks to be instantly transparent. As Jordà writes, 'highly idiosyncratic instruments which are often used only by their creators may not be the best sign or strategy for a serious evolution in this field' [10, p. 326].

It has a chicken and egg problem with ubiquity. Audiences cannot appreciate the music a new instrument makes because they have not developed an understanding of how it works. But audiences will not develop this understanding because they do not appreciate its music and so do not gain exposure to it. Being consistent (i.e. predictable once the pattern is learnt) and drawing as far as possible upon embodied metaphors (i.e. formal conventions) assist it in overcoming this initial hurdle. The instrument as tool allows new forms of musical expression, and it is this new music that excites and challenges its audience and creators.

The instrument as a composition

But the instrument as part of a musical composition is an art form in itself, challenging the audience's ideas of what an instrument is, what musical performance is. It is learning about the instrument itself and how it relates to the sound it is producing that is engaging its audience. It may ground itself in formal conventions initially, but then challenge the audience to keep up as it breaks them. In particular, however, the relationship between instrument and audience *develops* throughout a performance. The audience is not just learning the developing structure of the music, but the developing structure of the instrument: its sounds, mappings and possible types of interaction.

There is clearly a conflict between these two types of instrument. Is there a middle path? Certainly, one can write a composition specifically to showcase and teach an audience about a new instrument, an approach common within the NIME community. However, when considering what drives us to build instruments and see them in performance, we believe the most exciting and unexplored direction for the future lies in pushing this latter type of instrument towards the extreme. The fact an instrument is completely novel to an audience is not a hindrance to be overcome. It is the reason we are there to see it! Each new instrument is a unique interpretation of how action connects to sound and understanding this interpretation can be as aesthetically rewarding an experience as listening to the sound itself.

3. COMPOSING INSTRUMENTS AND PERFORMING MAPPINGS

Magnusson [12] discusses 'composing an instrument' as defining and limiting the boundaries of a musical space to be traversed in performance. We expand on this idea and propose an approach to instrument creation as an art form in itself where instrument, mapping and music are an integrated part of a greater composition. On the surface, this involves the music, mapping, gesture, physical form and performance space all being constructed around and supporting the same narrative. But just as a composer will carefully consider how far each musical idea may be exposed at each point of a composition, the workings and possibilities of a novel instrument should not be revealed in an unconsidered way. The exposition of the instrument, its range of interactions and sounds, the performer's gestures, are as much a temporal art form as the music itself. The mapping is *performed*: interactions are expressively presented and developed, each coherently building on what has preceded it. The structures of each aspect of the greater composition emerge simultaneously and in sympathy with each other.

3.1 On interactive composition

Interweaving instrument and composition is, of course, not new. *Interactive composition* was proposed by Chadabe [2] as a performance process where a composition programmed generatively within a system is unleashed in performance (usually by the composer). However, this approach still regards the instrument as a somewhat static tool that allows the purely musical ideas of its creator to be expressed. Our approach considers instrument and music as mutually dependent parts of unified composition, with their relationship explored and developed within a performance. The

instrument itself becomes a temporal art form.

An instrument built around playing a specific composition may be criticised for its inability to play a diversity of pieces. Jordà argues that 'a highly sophisticated "instrument" with a low [diversity of pieces that may be played on it] may be a very good interactive composition, but should not be considered as an instrument, even if it comes bundled with a hardware controller' [10, p. 335].

There is of course a distinction missing here as to whether this 'bundled controller' is to be handed to the listener, who most likely has never seen it before, or whether it is to be used in performance on a stage by a rehearsed and experienced musician. But beyond arguing over definitions, this prescribed delimitation overlooks an important possibility: that it is precisely within such a tightly constrained domain that new ideas happen [17], new ways of using (and abusing) an instrument are found [8], and new compositions, or even new types of music, are created. In a time when musical programming languages have unleashed a bewildering amount of sonic potential, it is the constraints rather than the affordances of an instrument that characterise it [12].

3.2 Design principles

Our approach suggests the following design principles.

Principle 1. *Design for a single performance*

The main consequence of this attitude towards making new instruments is a greater focus on performance – a single performance – during the design and creation of an instrument. This not only involves letting music and instrument be mutually influential as they are created, but thinking about how they will be presented together, their combined impact on the audience, how this relates to the character and narrative of the performance. Beginning by developing the concepts and themes behind the performance is an effective way to achieve this.

Principle 2. *Consider the rate that structures emerge*

For structures to be *continually* emerging, a careful balance of affirming expectations and creating surprises is necessary to allow close consideration of the amount of learning demanded of the audience at any given moment. Thus, the workings of the instrument should develop *throughout* the performance *together* with the music. Exposing the entire instrument at the beginning and then moving on to the 'real music' not only makes maintain a coherent narrative difficult, but also demands a shift in the audience's frame of mind during the piece.

Principle 3. *It is easier to begin 'in the dark'*

Careful consideration should be given as to what information is imparted before the performance. A preceding talk or programme notes explaining how everything works may be the right decision, as having the themes within a piece of music explained prior to listening can sometimes make it easier to appreciate – but not always. An audience who have no idea what to expect, what the limits of the performance are, is a gift only those musicians with novel instruments have. It is to be exploited rather than remedied.

4. EXAMPLE: THE SERENDIPTICHORD

The Serendiptichord is a wearable instrument for dancers, the result of a collaboration between artist Di Mainstone (the second author) and sonic interaction researcher Tim Murray-Browne (the first author). Combining ideas from musical interaction with Mainstone's sculptural, 'body-centric' work, it is both instrument and the performance and narrative in which it features. We describe it here to illustrate how the above principles may be applied.

Figure 1: Heidi Buehler with the Serendiptichord in performance at the *ACM Creativity & Cognition Conference 2009*. Photo: Deirdre McCarthy

The creation of the Serendiptichord began by developing the concept behind it and the ideas it embodied: exploration, discovery, serendipity, inspiring creative movement and provoking playful behaviour (**Principle 1**). These themes informed every aspect of the instrument: its shape, the sounds it makes, how the dancer interacts with it, the way these interactions create sound, and how it is introduced in a performance (**Principle 1**). As the instrument was developed, a narrative emerged of the relationship between performer and instrument through stages of discovering the instrument, playfully exploring how it may connect to her body, becoming gradually more sinister as it begins to possess and dominate her, reaching a climax whereupon she tears it off herself, and finally a return to the innocence of before as she resists its attempts to entice her once more. The narrative not only serves to unify music, instrument and interaction: it provides a framework for the instrument to be communicated to the audience. Whilst the dancer has rehearsed extensively, her journey of discovery allows the audience to discover its facets, capabilities and personality vicariously (**Principle 2**). The Serendiptichord is not demonstrated before a performance, nor does its shape communicate how it works. Establishing transparency – the connection between audience and instrument – is part of the aesthetic experience (**Principles 2 and 3**).

The instrument is made up of a headpiece module that rests on the shoulders and extends over and in front of the head, and two hand-held modules that may be attached to the headpiece or other parts of the body (Figure 1). With an exterior of only wood and red leather, but a form inspired by the curvaceous nature of acoustic instruments, it is shaped to be elusive but enticing (**Principle 3**). Hidden inside are four wireless accelerometers. Two of these, in the left-hand module and behind the neck, use a mapping metaphor [18] of a percussive instrument with sampled sounds modelled as spheres positioned within their orientation space. They are triggered when the dancer rotates the sensors to 'hit into' the sounds, with the speed of movement mapped to the volume of the sample. This mapping is created to be *expressively* transparent: left still it is silent but the more aggressively it is moved the more aggressive its sound be-

comes. Each sample is routed through a distinct effects rack controlled by an 'intensity' parameter. The intensity of the nearest triggered sample rapidly increases when the right-hand pod is shaken and slowly decays over time. The final accelerometer controls a frequency-shifting effect applied to the master channel. Embedded within the 'trunk' of the headpiece, which swings from side-to-side, it creates the sounds most characteristic of the instrument and is the most transparent part of the mapping with its continuous connection between frequency and orientation. More detail about the mapping may be found in [13].

The narrative of a performance is divided into chapters specifying the character of the instrument, the nature of the relationship between dancer and instrument and which modules are used, as well as which samples are available to be triggered (controlled back-stage). These provide a wide scope for improvisation, but allow control over how and when different facets of the instrument are exposed, which we describe in more detail to show how **Principle 2** may be applied in practice. In a typical ten minute performance, only the box housing the instrument is visible for the first minute as the dancer creates anticipation of its contents (**Principle 3**). Once opened, just the left-hand module is revealed. We quickly realise that movement of it causes sound but it is not yet clear how they relate. The dancer emulates our limited understanding and explores this relationship. The nearly identical right-hand module follows and we might expect, as the dancer apparently does, that it behaves in a similar fashion. But our expectations are not met, and we learn that shaking it intensifies the sounds triggered by the left hand. The headpiece – perhaps the most distinctive part of the Serendiptichord – is not revealed until around a third of the way through the performance. When the nature of the entire mapping has been established, its limits are explored as the performance turns more macabre and chaotic through to a climax and recapitulation.

The Serendiptichord has been very well received by audiences with invited performances at the Barbican, the Victoria and Albert Museum and Kinetica Art Fair in London. Comments from audience members suggest part of the 'hook' of the performance is from raising the question: *is the instrument really making the sound or is it prerecorded?* It is initially unclear, and seeking the answer motivates the audience to understand the connection between interaction and sound (**Principle 3**). As the show progresses the instrument becomes more transparent, with the direct mapping of the trunk irrefutably connecting movement and sound (**Principle 2**).

5. DISCUSSION

Novel instruments created both as a tool and the artistic output of their creators suffer a conflict: simultaneously learning how they work and appreciating their full complexity can be overwhelming. It is our hope that awareness of this issue will be liberating rather than off-putting. By following the path of instrument as art form, the instrument itself becomes a part of the composition. It does not need to conform to traditional modes of learning and performing and it may be quite idiosyncratic. The model of musical appreciation discussed is of course highly simplified but it suggests that the amount of learning demanded of the listener at each moment throughout a performance is an important metric to consider – one that also arises in an information theoretic analysis of music [1]. This can be done more effectively if all of the different aspects of instrument, music and performance are composed together cohesively.

Our approach focused on audience members who have not previously seen the instrument performed. It will be of interest to develop it to include those who have. Whilst many enjoy hearing the same piece of music performed again, is the same true of composed instruments? Do following our principles make this more likely or less? Furthermore, how may principles such as these be objectively evaluated?

Finally, it should be reiterated that an instrument created around a single performance piece need not be restricted to playing only that. It is around the constraints of such a space that creativity happens. Building an instrument around the musical space implied by one composition and then exploring its limits allows those ideas to be adapted and developed into something new in a way that the blank canvas of a limitless 'super instrument' perhaps does not.

6. ACKNOWLEDGEMENTS

This research was funded by a Doctoral Training Account from the Engineering and Physical Sciences Research Council (UK). The Serendiptichord was originally commissioned by the Centre for Digital Music under the Platform Grant (EPSRC EP/E045235 /1) for the *ACM Creativity and Cognition Conference 2009*, produced by BigDog Interactive and supported by the Interactional Sound and Music Group. Special thanks to Rachel Lamb, Judy Zhang, Stacey Grant and Vesselin Iordanov for their assistance.

7. REFERENCES

[1] S. Abdallah and M. D. Plumbley. Information dynamics: patterns of expectation and surprise in the perception of music. *Connection Science*, 21(2):89–117, 2009.

[2] J. Chadabe. Interactive composing: an overview. *Computer Music Journal*, 8(1):22–27, 1984.

[3] P. Cook. Principles for designing computer music controllers. In *Proc. NIME'01*, 2001.

[4] M. Csikszentmihalyi. *Flow: The psychology of optimal experience*. Harper & Row, New York, 1990.

[5] S. Fels. Designing for intimacy: Creating new interfaces for musical expression. *Proc. IEEE*, 92(4):672–685, 2004.

[6] S. Fels, A. Gadd, and A. Mulder. Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound*, 7(2):109–126, 2002.

[7] M. Gurevich, P. Stapleton, and P. Bennett. Designing for style in new musical interactions. In *Proc. NIME'09*, 2009.

[8] M. Gurevich, P. Stapleton, and A. Marquez-Borbon. Style and constraint in electronic musical instruments. In *Proc. NIME'10*, 2010.

[9] D. B. Huron. *Sweet anticipation: Music and the psychology of expectation*. MIT Press, 2006.

[10] S. Jordà. Instruments and players: Some thoughts on digital lutherie. *JNMR*, 33(3):321–341, 2004.

[11] T. Machover. Instruments, interactivity, and inevitability. In *Proc. NIME'02*, 2002.

[12] T. Magnusson. Designing constraints: Composing and performing with digital musical systems. *Computer Music Journal*, 34(4):62–73, 2010.

[13] T. Murray-Browne, D. Mainstone, N. Bryan-Kinns, and M. D. Plumbley. The Serendiptichord: A wearable instrument for contemporary dance performance. In *Proc. 128th AES Convention*, 2010.

[14] E. Narmour. *The analysis and cognition of melodic complexity: The implication-realization model*. University of Chicago Press, 1992.

[15] S. Nicolls. Seeking out the spaces between: Using improvisation in collaborative composition with interactive technology. *Leonardo Music Journal*, 20:47–55, 2010.

[16] J. L. Sherry. Flow and media enjoyment. *Communication Theory*, 14(4):328–347, 2004.

[17] P. D. Stokes. *Creativity From Constraints: The Psychology of Breakthrough*. Springer, 2006.

[18] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

Clothesline as a Metaphor for a Musical Interface

Seunghun Kim, Luke Keunhyung Kim, Songhee Jeong, Woon Seung Yeo
Audio & Interactive Media Lab
Graduate School of Culture Technology, KAIST
291 Daehak-ro, Yuseong-gu, Daejeon,
Republic of Korea
{ seunghun.kim, dilu, dearestj }@kaist.ac.kr, woon@kaist.edu

ABSTRACT

In this paper, we discuss the use of the clothesline as a metaphor for designing a musical interface called Airer Choir. This interactive installation is based on the function of an ordinary object that is not a traditional instrument, and hanging articles of clothing is literally the gesture to use the interface. Based on this metaphor, a musical interface with high transparency was designed. Using the metaphor, we explored the possibilities for recognizing of input gestures and creating sonic events by mapping data to sound. Thus, four different types of Airer Choir were developed. By classifying the interfaces, we concluded that various musical expressions are possible by using the same metaphor.

Keywords

musical interface, metaphor, clothesline installation

1. INTRODUCTION

Airer Choir is an interactive "clothesline" installation in which the metaphoric action of hanging clothes on a clothesline or clothes airer is recognized as creating sonic events. Thus, people can participate in musical performance by hanging "clothes". The aim of the installation is to determine the potential of an ordinary everyday action to represent, with the help of digital technology, an artistic idea.

There have been many works in which sonic events are created by specific objects, but the latter have been limited by their attachment of markers or electronic circuits. For example, in exhale[6], the clothes are made of conductive fabric used for interaction with the users. In Flock[4], audience can participate in the work by wearing the hat attached with a white LED. However, Airer Choir is a unique interactive installation because the clothing of anyone can be a medium for participation in the work.

This work is developed as a part of the Simple and Easy-to-use Musical Interface (SEMI) project. The SEMI project aims to design musical interfaces which provide easy-to-use control, facilitate multimedia performance, and offer enjoyable experiences for diverse audiences[5].

Based on the clothesline metaphor, four different types of musical interface that were developed for the performances and exhibitions are introduced. We propose how the mappings were applied differently in the four different versions.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. CLOTHESLINE INTERFACE

2.1 Interface based on an ordinary object

The musical interface as "clothesline" was designed to convert one of our daily tasks into a musical expression. Thus, an ordinary object was modified for musical expression.

Many musical interfaces were based on ordinary objects. In particular, some interfaces focused on a ball metaphor because the ball is simple but familiar in virtually all cultures and various gestures can be used to control the interfaces. For example, in BRBI[10], rotation, spin, shake, and throw gestures can be measured. In Twinkball[9], MIDI notes are generated based on the gestures of grasping, shaking, and approaching the light. In another case, a virtual ball object is used for bouncing and passing by performers who have smartphones[3].

Tanaka[7] argued that these types of musical interface have forms independent from existing musical instruments, but they also can be designed as a model of an instrument. The four different types of Airer Choir also have this characteristic. They recognize the clothes hanging on the clothesline and sonic events are generated based on processed information about the clothes.

2.2 Clothesline as a metaphor

By an analogy with a familiar object in HCI, a metaphor is a guide to learning how to use an unfamiliar object[1]. Thus, various musical interfaces have been designed based on metaphors. Similarly, this work is also based on the hypothesis that a metaphor is a strategy to make interfaces more intuitive.

Compared with other common objects such as a ball, the clothesline metaphor induces users to play with the interface naturally. On the other hand, audiences have a problem playing with a ball-shaped interface without any instructions. They do not know they can grasp, shake, or even throw it because the ball was not designed originally as an object for musical expression and there are various existing gestures for using a ball. However, the clothesline metaphor limits the gesture to hanging clothes only.

In interface design, expressivity depends on transparency, which in this context means how the output of a device from an action corresponds with the expectation of both performer and audience[1]. For performers, the transparency depends on two facts: cognitive understanding and proficiency. Use of the clothesline metaphor is highly appreciated for the two facts. In terms of cognitive understanding, when considering a clothesline as a musical interface, anyone can easily understand that the action of hanging clothes is converted into a musical expression. In terms of proficiency, most people can enjoy the interface because hanging clothes is a very common action.

3. INTERFACE DESIGN

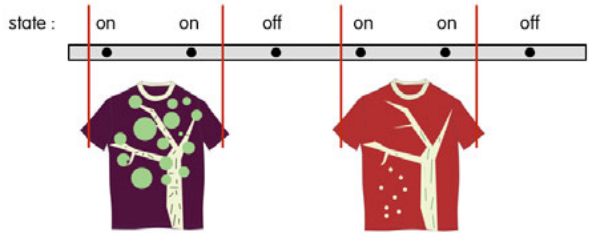


Figure 1: Several points store the data on the hanging clothes (discrete case)

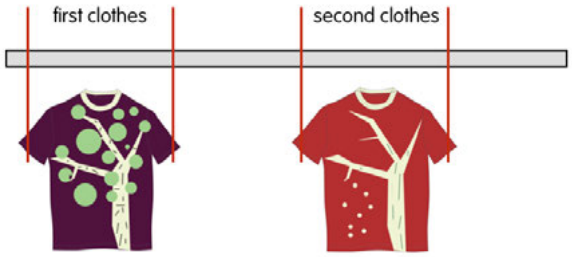


Figure 2: Ranges of clothes are stored (continuous case)

Based on the clothesline metaphor, because there may be various ways of input and output in the design of the interface, we discuss what kind of input and output is appropriate in this section. Not everything proposed here is applicable to the four individual interfaces because of the purpose and environment of each, but the followings are necessary for the ultimately ideal musical interface.

3.1 Input

In the clothesline interface, input can be a form of recognizing the gesture of hanging clothes itself or recognizing the clothes hung on the clothesline or clothes airer. In this work, we chose the latter because in order to use any article of clothing as a medium to interact with the interface, sensors and electronic circuits should not be attached on the clothes. In this case, recognizing the clothes is easier than detecting the movement of the body.

In order to measure how the clothes are hung, each sensor can be placed at regular intervals. In this way, there are a limited number of points along the clothesline that store the data on the hanging clothes (discrete case, Fig. 1). In contrast, ranges of clothes hanging on the line can be stored by using cameras (continuous case, Fig. 2). The former way can be implemented easily and clearly, but it does not separate the clothing and only determines the status of each point. However, in the continuous case, several virtual clothes objects can be created in the program. Creating individual sonic events becomes possible when ranges of the clothes items are stored.

One type of input for the points, or virtual clothes objects, along the clothesline is Boolean variable storing in an on/off state to judge whether the clothes are indeed hung. Another type of input is a continuous number that represents a unique characteristic of the clothes such as color or the state of the hanging clothes such as position, movement, and pressure.

3.2 Output

Basically, each points, or virtual clothes object, along the clothesline creates individual MIDI notes or plays sound samples by responding to the input. Thus, the user can experience a new sonic event from the speakers near the hanging clothes.

In addition, even if the same clothes are hung, users can experience various sounds by creating a different sonic event as the position is changed from side to side. To achieve this variety, each point along the clothesline has its own unique sound sample. Alternatively, a sound is also changed by controlling the volume balance between the loudspeakers as the position of the clothes.

A unique characteristic or the state of hanging clothes is used for a parameter of the MIDI note or the sound sample. This input varies the sonic event when the hanging clothes are different colors or when they are hung tightly or loosely. Thus, inputing unique characteristics creates a dynamic user experience.

According to the mapping types defined by Tanaka[8], the creation of a sound sample by recognizing an item is considered to be binary mapping. The number of MIDI notes or the balance between the speakers, which represents the position of the clothes, is considered to be basic parametric mapping. Changes in the parameters of the sound effects, which represent the characteristics of the clothes hung on the clothesline interface, can be considered to be expressive mapping. Although all implemented clothesline interfaces are not the same, the mapping strategy designed for the interface is complex mapping, which means that a single input is used for various output events.

4. IMPLEMENTATION

4.1 First work

The first clothesline interface was implemented for a simple demonstration in a short indoor performance (Fig. 3). In the performance, two 5m ropes were tied between two pillars to create a clothesline. By hanging and gathering several items of clothing, the creation of various sonic events was represented. Visual effects were also displayed over the hanging clothes by a projector.

In order to recognize the clothes, eight photo sensors were placed on two lines. First, the intensity of light without the hanging clothes was measured by each sensor because a change in light indicates a change in state (i.e., clothes are hanging or gathered). When latter decreases below the threshold, this system indicates that an item of clothing is hanging on the clothesline. In contrast, it indicates that the clothes are gathered when the value increases above the threshold. Moreover, by separating the threshold into several levels, we tried a kind of basic parameter mapping by creating different sonic events when the number of clothes hanging on the sample position was different.

We tried both ways of the sound output discussed above: MIDI note and sound sample. Because each photo sensor relates to its own MIDI note or sound sample, a different sonic event is created when the position of the clothes is changed.

4.2 Second work

The second interface is a large-scale interactive installation that can be installed both indoors and outdoors. Audiences can participate together in the performance by hanging their own clothes on a long clothesline. Thus, the characteristics of the second clothesline interface are similar to Soundnet[7], which emphasizes the quality of large scale and multiuser instrument.

This work was installed on the lawn at KAIST. A 50m

Figure 3: Performance with the first interface

Figure 4: The sonic event is expressed through the loudspeakers near the hanging clothes

line was placed and PVC pipes were used as columns to bear the long rope. Behind the line, there were several cameras to track the items of clothing hung on the clothesline, the computers, and the loudspeakers. Audiences could freely hang and gather clothes on the line. The response to hanging the clothes was expressed by the sonic event of a bell sound through the loudspeakers around the clothes (Fig. 4). The generated bell sound depended on the color, position, and sway of the clothes.

In the recognition process, the cameras decided whether or not the clothes were hanging and measured their position, color, and sway. These measurements were used as parameters for sound synthesis. The color decided the type of bell sound. The clothing position controlled the volume balance between the loudspeakers so that the audience hanging the clothes could hear the loudest bell sound. The movement value was utilized for reverberation effect, so the more a piece of clothing swayed, the stronger a reverberation effect occurred.

4.3 Third work

The third Airer Choir was developed for Incheon Digital Art Festival (INDAF), which exhibited interactive installations and art works during a month-long exhibit. Because the interface was required to be exhibited for public over a long period, durability was an important difference from the second work. Thus, although the metaphor of hanging clothes was still used for the work, pipes are used instead of rope as a visual representation of the work.

In the recognition process, a new method was used. Several tone holes were drilled along the pipes, a speaker unit was placed at one end of the pipe, and a small microphone was placed at the other end of the pipe. The signal from

Figure 5: The third Airer Choir interface exhibited in INDAF

Figure 6: The fourth Airer Choir interface exhibited in Studio SEMI

the microphone was filtered, amplified, and played by the loudspeakers. As a result, the resonant sound in the pipe depended on how the tone holes were covered by the clothes.

This work was installed in an outdoor exhibition space (Fig. 5). Four 2m pipes with speaker units and microphones at each end were suspended in the air. The resonance in the pipe was also played by the loudspeakers installed on the floor so that audiences could experience the sonic event by hanging clothes on the pipes.

4.4 Fourth work

The fourth interface was developed for Studio SEMI, which exhibited the musical interfaces developed by the SEMI project during a period of two weeks. This exhibition focused on interaction with the audience, so durability was also important. Therefore, the overall shape of the work was similar to the third work as shown in Fig. 6.

However, we applied a different mapping strategy. A piezo sensor to measure the vibration was placed in the middle of a pipe. Thus, a sonic event of synthesizing a sound was generated when there was a vibration in the pipe, and the magnitude of the vibration was used for the reverberation effect and volume parameter. A difference from previous works was that this work recognized the pressure on the pipe created by hanging and gathering the clothes. The sound reverberated strongly when the clothes were placed firmly on the airer interface. The reason we used this mapping strategy was that because this work was installed in an indoor space, the continuous sound may have interfered with other works if is was played continuously due to the hanging of the clothes.

5. DISCUSSION

System	Focus	Media	Scale	Musical Ranges / Notes	Sensor	Directed Interaction	Mapping
First work	player	sound, image	1	limited, loops	photo sensor	no	binary(on/off), basic parameter(loop)
Second work	audience	sound	1-10	loops, delay, panning	camera	med-high	binary(on/off), basic parameter(loops), expressive(delay, panning)
Third work	audience	sound	1-5	pitch	microphone	low	basic parameter(pitch)
Fourth work	player, audience	sound, image	1-5	loops, delay	piezo sensor	low	binary(on/off), expressive(delay)

Table 1: Contexts of the musical interfaces using the clothesline metaphor

We classified the four interfaces discussed above as the criteria proposed by Blain[2] with the mapping strategy criteria proposed by Tanaka[8]. Because the four works use the same metaphor, some elements are identical. The common purpose of these works is to design an easy-to-use interface using a familiar object, so the learning curve is steep and there is no pathway to expert performance. Other elements demonstrating the difference between the interfaces are shown in Table 1.

By applying the variables differently according to the purpose and environment, we verified that the same metaphor could represent various musical expressions. Also, the works were designed to have high transparency because the clothesline metaphor is based on an ordinary object that is familiar to most people, and it indicates that hanging clothes is the gesture for interaction. Thus, audiences enjoyed the works easily without any instructions in the exhibitions (A video sample is available at **http://aimlab.kaist.ac.kr/∼asuramk88/clothesline**).

However, several limits of the metaphor exist. Because a fast interaction is impossible with this type of work, the performance may be lackluster without an ancillary element such as background music. In addition, ease of learning means that there is almost no difference between expert and novice, so the level of play may cause users to lose interest quickly.

6. REFERENCES

[1] A. Blackwell. The reification of metaphor as a design tool. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 13:490–530, Dec. 2006.

[2] T. Blaine and F. S. Contexts of collaborative musical experiences. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, pages 129 – 133, Montreal, Canada, 2003.

[3] L. Dahl and G. Wang. Sound bounce: Physical metaphors in designing mobile music performance. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, pages 178 – 181, Sydney, Australia, 2010.

[4] J. Freeman and M. Godfrey. Technology, real-time notation, and audience participation in Flock. In *Proceedings of the International Computer Music Conference*, 2008.

[5] M. Hur. SEMI: simple easy-to-use musical interfaces. Master's thesis, KAIST, 2009.

[6] T. Schiphorst. exhale: breath between bodies. In *Proceedings of the ACM SIGGRAPH 2005 electronic art and animation catalog*, pages 62–63, New York, NY, USA, 2005. ACM.

[7] A. Tanaka. Musical performance practice on sensor-based instruments. In *Trends in Gestural Control of Music*, pages 389 – 406. Ircam - Centre Pompidou, 2000.

[8] A. Tanaka. Mapping out instruments, affordances, and mobiles. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, pages 88 – 93, Sydney, Australia, 2010.

[9] T. Yamaguchi, T. Kobayashi, A. Ariga, and S. Hashimoto. TwinkleBall: a wireless musical interface for embodied sound media. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, pages 116–119, Sydney, Australia, 2010.

[10] W. Yeo. The bluetooth radio ball interface (BRBI): a wireless interface for music/sound control and motion sonification. In *Proceedings of the International Computer Music Conference*, New Orleans, USA, 2006.

EGGS in Action

Pietro Polotti
Department of New Musical Technologies and Languages
Conservatory of Music G. Tartini
Via Ghega, 12, Trieste, Italy
Interaction Group
IUAV, University of Venice, Italy
pietro.polotti@conts.it

Maurizio Goina
Department of New Musical Technologies and Languages
Conservatory of Music G. Tartini
Via Ghega, 12, Trieste, Italy
maurizio@goina.it

ABSTRACT
In this paper, we discuss the results obtained by means of the EGGS (Elementary Gestalts for Gesture Sonification) system in terms of artistic realizations. EGGS was introduced in a previous edition of this conference. The works presented include interactive installations in the form of public art and interactive onstage performances. In all of the works, the EGGS principles of simplicity based on the correspondence between elementary sonic and movement units, and of organicity between sound and gesture are applied. Indeed, we study both sound as a means for gesture representation and gesture as embodiment of sound. These principles constitute our guidelines for the investigation of the bidirectional relationship between sound and body expression with various strategies involving both educated and non-educated executors.

Keywords
Gesture sonification, Interactive performance, Public art.

1. INTRODUCTION
In this paper, we report three years of explorative work about gesture sonification on the basis of the EGGS system introduced in NIME08 [5]. We present various realizations both in the form of public installations and interactive performances. The main principle stated in 2008 is respected in every work: elementary sounds are defined and employed for the sonification of a small number of gesture categories. Roughly speaking, these categories can be subdivided into the two main classes of straight and circular movements. In all of the cases, we defined various and alternative instances of elementary gestalts in terms of sonification sound sets, intended as unitary cognitive structures activated by gestures according to the particular goal and context of the artistic realization. We also realized extensions of the principles to the visual domain in a multimodal sense. This is also meant as a possible way for defining effective indirect mapping between visual forms and sounds with gesture as a sort of interpreter.

As already illustrated in [5], we think of gesture as generated by sequences of points forming trajectories in space that can be combined in order to generate complex cognitive objects at different possible levels. Our approach to gesture analysis is, thus, essentially abstract, different from an expressive/emotional feature extraction as, for example, in the work by Camurri and al. [2]. Expressiveness is a consequence of the combination of sonic/cinematic relationship in articulated structures jointly to the effect of further parameters such as sonic/gesture dynamics relationships. In this sense, one of the sources of inspiration of this work are the theories of Paul Klee. In particular, we inherit from Klee the idea that a dot is the ur-element that is the atomic element, whose movement generates lines and planes. This concept is well illustrated in Klee's "Pedagogical Sketchbook" [8], a book intended as the basis for the course in Design Theory at Bauhaus.

The paper structure is the following. The next section provides a brief review of the EGGS principles and system. Section 3 describes the public art realizations. Section 4 illustrates the interactive performance works. In Section 5, we discuss possible future developments and applications, and draw our conclusions.

2. THE EGGS SYSTEM
EGGS provides a basic system for gesture sonification. Arbitrary articulation and combination of elementary mappings can be defined. The system allows applications for performing arts and interactive dance as well as for public interactive installations. Differently from what pursued in other works on gesture-sound mapping [1], [10], we opt for a simple cinematic and dynamic gesture analysis valid for general categories characterized by some abstract and elementary properties. In order to do this, we consider gestures as spatial trajectories of moving points, the point being a hand, an elbow or a knee, and we define segmentation criteria based on simple geometric considerations. In EGGS, visual data of a point-wise source are processed by a trajectory tracking routine that returns different indexes corresponding to five categories: straight movements, circular clock wise and circular counter clock wise movements, direction inversion and stillness (see [5] for more details). A variety of second order parameters are derived from the primary data related

NIME'11, May, 30- June 1, 2011, Oslo, Norway

to the purely cinematic trajectory that is the movement scalar and vector velocities and accelerations. Up to now only a 2D tracking was taken into consideration.

Using the EGGS system, as in any musical practice, involves learnability issues (see, for example, [7] for a discussion about apprenticeship in new musical interfaces). Exercise is important in order to understand the possibilities of the instrument and obtain relevant results. This is the case, when working with professional performers. However, the system is immediate and easily usable by anybody as any simple gesture produces a meaningful sonification. The latter is the approach taken into consideration in case of public installations, where the visitors should be immediately able to get a satisfying, enjoyable and stimulating feedback from the system. Indeed, differently from the perspective introduced in a recent work on interactive dance [9], our work is intended to spur the performer (or user) to control, adapt and explore her/his gesture according to the received continuous sonic feedback in an enactive way.

Given these founding criteria, the fashion we carried on our investigation and development of the system was in spirit of experimentation of multiple alternative realizations of the same principles in different artistic applications. We consider this way of proceeding to be in analogy to a design practice, in which the following realization depends on the previous results in a cyclic way and a comparative evaluation of the different realizations is a matter of enrichment and deepening of knowledge about some subject; in our case the effectiveness of sound as continuous feedback for controlling gesture and the expressivity of the sound-gesture couple in a multimodal sense. The latter point is considered both from the spectator point of view (audio and visual multimodal aspects) and from the actor point of view (audio and proprioceptive multimodal aspects).

An extension of the system to the case of a multimodal feedback was also considered. We investigated the possibility offered by using gesture as a control of both sound and graphic generation, where the correspondence between sounds and images was in general arbitrary. In this sense, we refer to Chion's definition of audiovision [3], according to which any association of images in movement and audio produces a composite object that lives in a third dimension, which is multimodal: a sort of vector product. On the other side, a concurrent generation of sound and graphics by means of the same movement analysis allows to search for novel relations between sound and image through the juxtaposition of abstract and elementary categories "unified" by the same gesture.

3. INTERACTIVE INSTALLATIONS IN A PUBLIC ART FASHION

Visual Sonic Enaction (VSE) is a multimodal and interactive installation that allows to generate an audiovisual representation of one's gestural expressivity. VSE is usually presented in a ludic fashion, by introducing it to the public through the metaphor of graffiti painting: the visitors are encouraged to paint on a large wall by means of an "electric torch/spray can" controlling different graphic and sound processing algorithms. The sound elicits and guides the movements of the user and immerges she/him in a bodily-visual-auditive experience, by producing a multimodal and continuous feedback to the gesture. Indeed, sound plays the role of connective element of the three components of VSE.

In the current version [14], three sets of elementary sounds and three groups of simple graphic signs were defined and employed for the sonification and visualization of two basic main categories of gestures/movements: straight and circular. The visitor could experiment only one sonic set at a time for each visual-sonic self-portrait. On the contrary, in the context of a single portrait, the user could change graphic set in any moment and any number of times, by shaking one of three coloured bottles put aside the interactive area. The bottles were equipped with wireless accelerometer sensors. The three graphic typologies are illustrated schematically in the bottom-right corner of Figure 1. The three kinds of sounds were experimented one after the other in three different visual-sonic portraits. The three sound sets, represented iconically in the top-right corner of Figure 1, included swishing and metallic sounds, low pitch FM synthesis sounds and glass and crystal tinning sounds generated by the physics-based sound synthesis package Sound Design Toolkit (SDT) [4]. The sound typology changed only when the user decided to save the current portrait and to start another one from blank. Different kinds of mappings were implemented, more or less variable in timbre and other characteristics in relations to gesture categorization. Some mappings implemented a discrete separation between straight and circular movements and other were modulated continuously with the curvature of gesture, in order to obtain a certain variety in each one of the nine possible sonic-graphic combinations. An example of a portrait is reproduced in Figure 1.

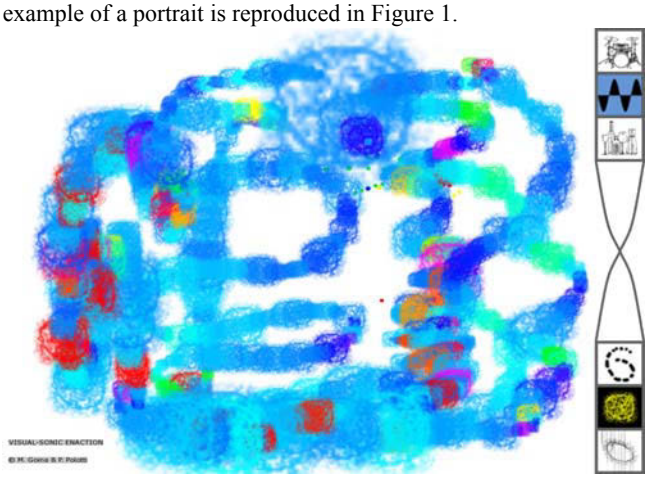


Figure 1. The VSE "canvas" with a visual-sonic self-portrait of one of the participants.

As a final product, the visitors received an audiovisual file as a record of the abstract visual-sonic representation of their gestural expressivity. Besides, anyone can watch and listen to the bodily expressivity of any visitor on the VSE website, where the "visual-sonic self-portraits" are uploaded. Someone among the visitors was able to interpret very quickly the spirit of the installation and adapted her/his gesture to the different sonic/graphic combination in order to reach a coherence of all of the three aspects involved in the installation (see for example Serena's portraits [14]).

In VSE, the EGGS principles are applied to the visual domain as well. The aim is not to paint. Rather, what appears on the wall or on the computer screen is a visualization of the expressivity of gesture. At the same time, in an enactive way, the visual feedback spurs the user to modify and control her/his own gesture also

according to different type of visualized graphic. The use of different graphic types is fundamental in order to uncouple the gesture from the idea of painting as well as from a particular sonic set. In the future, another aspect we want to investigate is if the definition of abstract (gestural) categories and the definition of effective, however independent, mappings for sound and graphics generation could reveal unexpected relations between images and sounds.

In another recent public installation, Sonic Walking (SW), we concentrated on gait expressiveness [13], therefore, shifting the focus from the upper part of the body to the lower part and from a creative to an everyday context, where a visitor has just to walk freely along a straight path in an ordinary indoor space. The gait of the visitors is sonified by means of ecological sounds related to nature and, more specifically, to the four basic elements fire, earth, air and water. In particular, we employed the sound of a big fire in a forest, the sound of a rain stick recalling that of sand or grains and the sound of a strong wind. The water had two versions: quite waves on a beach and a sound giving the impression of being underwater. The visitors experienced the five sounds in a fixed order: water, earth, fire, air, underwater.

Before starting to use the system, we told the visitor that they would listen to her/his footstep and that their footsteps would first dabble, then rustle, then crackle, then blow and, finally, go underwater. The visitors could walk along a path of approximately 8 meters and wore two lights tied to the external side of their knees, so that every light was detected by one of the two cameras located on the two side of the path. Also, the users wore wireless headphones in order to experience a more immersive and internal feeling of her/his body movements given by the continuous sonic feedback. The audio was as well reproduced by four loudspeakers located at the extremities of the path, so that the audience passing by could hear the gait sonifications. This case of EGGS application was the least articulated, since no cinematic analysis was taken into consideration, and only the dynamic aspects of movement drove the sonic feedback. A further development of SW, integrating visual elements and a 3D detection aiming at including circular trajectories, is previewed.

4. INTERACTIVE PERFORMANCES

In case of a stage context, working with a professional performer/dancer, sound is meant as an effect of the choreographic gesture and a representation of her/his gestural expressiveness. EGGS becomes what we could denote as a "choreophone": the performer/dancer does neither follow a musical piece, nor controls the execution of a musical piece, and not even generates any music with her/his movement [5]. Rather, (s)he listens to her/his gesture, enactively, modifying and controlling her/his performative action according to the produced sound. The sounds, thus, is a representation of the movement, a sonic consequence and a continuous feedback, in no way external to the gesture itself. In this fashion, sound is intended as augmenting the proprioception of the performer.

In the context of the latest SMC conference, we presented a performance entitled “Swish ‘n’ Break” [6]. The performance is conceived as a controlled improvisation on a predefined score of sounds and gestures, in the style of the previous performances realized by means of the EGGS system [12]. The sounds used in this performance are all derived from the Freesound project [11] and retrieved by means of a number of keywords defined in advance, in the spirit of a programmatic compositional approach. The keywords are: 1) Swish, 2) Nature (Air – Water – Fire – Earth), 3) Break. The choice of the keywords determines the overall structure of the performance, which is fixed and divided into three sections. The Nature section, the richest in sounds, is conceived as a gradual passage from a natural open-air soundscape to an indoor soundscape. Within each section, the sound-gesture mapping is configured according to the general principles of simplicity underlying the EGGS project, and based on a decomposition of the gesture cinematic into segments belonging to the already mentioned five categories: straight, clockwise, counterclockwise, direction inversion and stillness. A camera detects the bi-dimensional coordinates of two electric bulbs handled by the dance performer (see Figure 2). A correspondence between the dynamic of gesture and the dynamic and other parameters of sounds adds a further expressive layer. Within certain constraints, the live electronics players can change the mapping of the sound as well as the quality of the dynamic response of the system, engaging a dialogue with the performer/dancer. The final result is a performance based on a mostly predetermined sonic-choreographic score defined along a large number of rehearsals, in which the performer experiments how to create and adequate her/his gesture according to the system sonic feedback and, reciprocally, how to condition and control the sonic response by means of her/his gesture. All the choices in terms of evolution and refinement of the performance and of the EGGS system as a whole, are taken during the rehearsals as an agreement among all of the members of our group through discussions, trials, selections and optimizations of the gesture-sound mappings and their combination and concatenation. This corresponds to a creative methodology, where working in group and going through brainstorming and debate phases is a firm point.

In NIME 2011, we are going to present a new performance, entitled “Body Jockey”. The idea is to introduce embodiment in club culture and musical styles. The technical setup is the same as in Swish ‘n’ Break. Part of the sounds employed has been retrieved from the Freesound project, another part has been composed by the authors. The trio of performers acts as if being in a DJ and VJ set. The dancer triggers and modulates sounds by mean of her body, while the laptop performers change sounds and mappings, as well as the quality of the dynamic response of the system. The result is a dialogue of the laptop performers with the dancer, who follow a predetermined score, however leaving space to a controlled improvisation. A graphical representation of sounds and mapping is projected on the screen in order to add a video layer to the multimodal experience of the performers and the audience. Through gesture sonification, music becomes embodied in the dancer herself, and this feeling is transmitted to the audience attending the performance as well – the purpose is to create an enhanced disco-club environment, where body, music and video are jointly engaged in the audience experience. This is also an attempt to provide in the future a version of EGGS not limited to trained dancers but available to everybody.

Figure 2 The performer in action.

The overall structure of the performance is fixed and divided in three sections. Mapping and sounds change in every section. In the first part metronome is fast and fixed. Sounds are percussive and rhythmically constrained. The most used elementary gesture is the trajectory inversion, employed to trigger sounds. The main rhythmic patterns are the usual dance even meters. The sound volume is fixed and dynamic changes are obtained by modulating the density of sound events. The second part is more free-style and based on long sound events that are not rhythmically constrained. Here, the usual types of EGGS elementary gestures that is the straight and circular trajectory elements are important and used to modulate the sound parameters. The third conclusive part is similar to the first one in a disco dance style.

5. CONCLUSIONS

In this paper, we presented a number of artistic implementations of the EGGS system. In all of the works, we applied the EGGS principles consisting in i) treating sound as a representation of gesture and ii) working with elementary cognitive and abstract units in terms of gesture analysis, segmentation and sonification. Such an approach has the advantage of enhancing sound embodiment in interactive installations and performances: the sound-gesture binomial is cognitively fully integrated in a multimodal sense, auditory/proprioceptive from the actor side, and auditory/visual from the spectator side. Also, the basic strategy adopted for the gesture analysis and sonification provides a clarity of interpretation of the system response both in case of a professional performer and of an ordinary user. Reproducibility and learnability, in fact, are part of the main issues EGGS aims at.

These aspects allow to envisage many other potential applications for the EGGS system in other fields where body movements and its control in time are crucial, ranging from sensory-motion recovery to musical pedagogy and others. Concerning the latter aspect, a research project is in progress involving experimentation in primary schools employing a version of the system adapted to pedagogical purposes.

6. ACKNOWLEDGMENTS

This project is funded by the *Servizio università, ricerca e innovazione of the Regione Friuli Venezia Giulia* and co-funded by the *Fondazione Cassa di Risparmio di Trieste*.

7. REFERENCES

[1] Bevilacqua, F., Müller, R. and Schnell, N., MnM: a Max/MSP mapping toolbox, *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression,* (*NIME05*)*,* Vancouver, Canada, 2005.

[2] Camurri, A., Lagerloef, I. and Volpe, G. Recognizing emotion from dance movement: Comparison of spectator recognition and automated techniques. *International Journal of Human-Computer Studies*, 59(1):213–225, July 2003.

[3] Chion, M. *Audio-Vision -- Sound on Screen*, New York, Columbia University Press, 1994 [Original edition: *L'audio-vision. Son et image au cinéma*. Editions Nathan, Paris, 1990].

[4] Delle Monache, S., Polotti, P. and Rocchesso, D. A Toolkit for Explorations in Sonic Interaction Design, *Proceedings of Audiomostly '10*, Pitea, Sweden. Sept., 15 – 17, 2010

[5] Goina, M. and Polotti, P. Elementary Gestalts for Gesture Sonification, *Proceedings of the 2008 International Conference on New Interfaces for Musical Expression* (*NIME–08*). Genova, Italy, pp. 150–153, 2008.

[6] Goina, M., Polotti, P. and Taylor, S. Swish & Break - Geschlagene-Natur, *Concert around Freesound, SMC 2010, 7th Sound and Music Computing Conference*, Universitat Pompeu Fabra, Sala Polivalent, Barcelona, Spain, 22 July 2010.

[7] Jordà S. Digital Instruments and Players: Part I – Efficiency and Apprenticeship, *Proceedings of the 2004 International Conference on New Interfaces for Musical Expression* (*NIME04*)*,* Hamamatsu, Japan, 2004.

[8] Klee, P. *Pedagogical Sketchbook,* trans. Sibyl Moholy-Nagy. Frederick A. Praeger, New York, 1965.

[9] Schacher, J. C. Motion to Gesture to Sound: Mapping for Interactive Dance, *Proceedings of the 2010 International Conference on New Interfaces for Musical Expression* (*NIME 2010*), Sydney, Australia, 2010.

[10] Van Nort, D. and Wanderley, M. The LoM Mapping Toolbox for Max/MSP/Jitter, *Proceedings of the 2006 International Computer Music Conference ICMC 06*, New Orleans, USA, 2006.

[11] www.freesound.org (Mar. 30, 2011).

[12] www.visualsonic.eu/eggs_in_action.html (Mar. 30, 2011).

[13] www.visualsonic.eu/sw.html (Mar. 30, 2011).

[14] www.visualsonic.eu/vse.html (Mar. 30, 2011).

A Reverberation Instrument Based on Perceptual Mapping

Berit Janssen
Studio for Electro-Instrumental Music (STEIM)
Achtergracht 19
Amsterdam, The Netherlands
berit@steim.nl

ABSTRACT

The present article describes a reverberation instrument which is based on cognitive categorization of reverberating spaces. Different techniques for artificial reverberation will be covered. A multidimensional scaling experiment was conducted on impulse responses in order to determine how humans acoustically perceive spatiality. This research seems to indicate that the perceptual dimensions are related to early energy decay and timbral qualities. These results are applied to a reverberation instrument based on delay lines. It can be contended that such an instrument can be controlled more intuitively than other delay line reverberation tools which often provide a confusing range of parameters which have a physical rather than perceptual meaning.

Keywords

Reverberation, perception, multidimensional scaling, mapping

1. INTRODUCTION

Reverberation is ubiquitous as an audio effect, yet its musical potential is seldom fully realized. Reverberation is more than just smearing a signal in time, it can also convey a context to sounds, which can be fascinating especially in electro-acoustic music. A reverberation instrument to investigate its potential is needed, which allows control of spatiality based on perceptive qualities, rather than a multitude of physical parameters. In this paper, different techniques for artificial reverberation will be weighed against each other. On the basis of psychoacoustic research, an instrument for intuitive control of reverberation will be proposed.

2. CHALLENGES IN ARTIFICIAL REVERBERATION

Reverberation can be considered as a series of repetitions of a signal, as caused by sound reflection from walls and obstacles. As such, there are various ways of artificially reverberating a signal, which can be roughly divided into impulse response and modelling techniques. For both approaches, there are various challenges to be considered.

2.1 Impulse responses

The way a space reacts to signals can be measured by recording its response to a short noise signal, or its response to a sine sweep over all frequencies. This so-called impulse response can be used for reverberation by convolving it with a target signal. Convolution, in the time domain a very computionally expensive process, can be performed in real-time by short-term Fourier transforms. This technique underlies reverberation software such as the SpaceDesigner in LogicPro, the IR plug-ins by Waves, or Altiverb by Audio Ease, to just name a few.

Even though the quality of convolution reverberation is excellent, there are only limited options to influence the reverberation other than choosing a different impulse response. If, for instance, a user selects an impulse response from a forest, then from a city scape, and then decides that the desired reverberation should lie somewhere in between, the impulse responses cannot be interpolated in a meaningful way. The only option is to browse through and try all impulse responses that might apply.

Techniques of artificial reverberation that do not sample, but model acoustic spaces provide more options to influence the reverberation by modifying the model, but they also have some shortcomings, as will be discussed below.

2.2 Modelling

The most realistic reverberation through modelling can be achieved by creating a virtual representation of the space, and simulating the reflection of sound from surfaces. Image source and raytracing methods have been applied successfully to this end. However, these algorithms require careful weighting between model complexity and computation cost. Moreover, outdoor spaces are hard to simulate, since it is hard to ascertain whether irregular objects at large distances still contribute to the reverberation, while the virtual space needs to be bounded at some point. Most importantly, however, the parameters that can be varied in the virtual spaces, such as absorption of the surface, or the size and dimensions of surface elements, are not correlated with perceptual parameters, so it is difficult to create a specific reverberation.

Other, statistical modelling techniques imitate reverberation through delay lines. In principle, this can lead to realistic reverberation. Especially when the diffuse tail end of reverberation is concerned, a dense exponentially decaying series of delays can be a good approximation. Common design elements such as feedback delay lines have a similar effect on the spectrum as interference through wall reflection: parts of the spectrum may show frequency dips due to phase cancellation. Other acoustic effects, such as audible echoes, can also be modelled through long delay times, but are very hard to combine realistically with shorter delay times to a natural sounding reverberation. In short, the

choice of filter coefficients remains "half an art and half a science" [9].

The most significant problem of delay line modelling is that the parameters of filter networks — delay time and delay gain — are not perceptual parameters. The delay time is often referred to as "room size" in the controls of reverberation tools. Yet even this is related more to a visual than an acoustic spatial impression.

As has been seen in both the impulse response and modelling approaches to artificial reverberation, a concept of how humans perceive and categorize acoustic spaces, according to which impulse responses could be organized, or parameters of artificial reverberation could be tuned, is missing. This missing link of the human perception of reverberation can only be bridged by psychoacoustic research.

3. PSYCHOACOUSTIC RESEARCH

Much psychoacoustic research on reverberation has been done since the 1960s. Beranek [2] established some categories for describing concert hall acoustics, such as intimacy, distortion, tonal balance, spaciousness, ensemble, reverberation, or the ratio of early reflections to the reverberant sound field. He rated a number of North American concert halls on the basis of these categories. After a factor analysis of the ratings and comparison with the recorded soundfields he found the time interval between the direct sound and the first reflection, which he calls initial time delay gap, to be the most significant factor. However, since these results are based on his individual judgement, the globality of his findings is unclear.

Gottlob and Siebrasse [7] used impulse response recordings with an artificial head in various European concert halls, which they presented pairwise to a number of participants and asked them to state their preference. A factor analysis of the responses led to four factors which were found to be related to reverberation time, definition, inter-aural correlation, and the ratio of early lateral to total early energy.

Wilkens and Lehmann [6] used a similar experimental setup, but asked participants to rate the concert space acoustics on scales of antonym pairs. The results were subjected to a factor analysis, the resulting three factors of which were found to be related to strength of sound, centre of gravity time, and the slope of the early decay time of different frequency bands.

Next to research on existing spaces, which is strikingly limited to concert halls, there is also some interesting investigations using synthesized spaces and reflections. Berkley [3] simulated spaces using image source algorithms, varying the reverberation time and source-receiver distance. He reverberated speech signals this way, and presented them pairwise to a number of participants, asking for difference judgements. Multidimensional scaling yielded a two-dimensional representation of the perceived differences, the axes of which he found related to reverberation time and spectral deviation, a measure for the roughness of the frequency response of the room.

Ando [1] synthesized sound fields with an array of speakers. He investigated which artificial reflection patterns would lead to specific acoustic impressions. He found that the most important parameters for reverberation perception are strength, the delay time of early reflections, the subsequent reverberation time, and the inter-aural cross correlation.

The existing research on the perception of reverberation grants a lot of essential insights, however, often the aim consists in finding design parameters for "good" acoustics supporting speech and music performances, and excluding therefore reverberances which show timbral or temporal irregularities, or perceptible echoes. With this focus on only a limited set of acoustic properties of spaces, the more general question as to how humans perceive acoustic spaces and cognitively organize them, cannot be answered.

4. PSYCHOACOUSTIC EXPERIMENT ON INDOOR AND OUTDOOR SPACES

A psychoacoustic experiment designed by the author [5] therefore included indoor and outdoor spaces, which showed colouration and echoes. Moreover, it was decided that a nonverbal way of judging reverberation was preferable over semantic approaches, since the phenomenon of reverberation can only be captured in language to a limited extent.

4.1 Experimental setup

Impulse responses from four indoor and five outdoor spaces were recorded. The impulse was generated by a starter gun and the impulse responses were recorded with an omnidirectional microphone. The resulting impulse responses were normalized and combined to pairs, which allowed for a comparison of every individual impulse response with every other sound. The resulting 36 pairs were presented to a total of 49 participants, who were listening to the sounds via a high fidelity sound system, in groups of nine to sixteen participants at the same time. The participants were instructed to rate the similarity of the sounds in each pair, on a 16-point scale ranging from similar to dissimilar.

4.2 Results

The multidimensional scaling analysis (ALSCAL) of the occurring judgements could be represented very well in a two-dimensional map (see Figure 1), which explained 98.2% of the variation, with Kruskal's stress value at s=0.06.

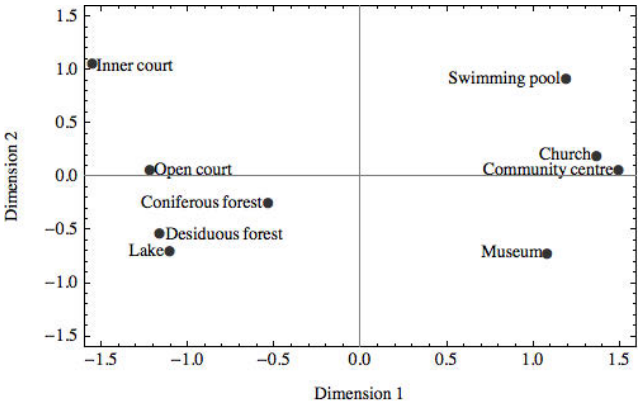


Figure 1: The two-dimensional map representing perceived differences of impulse responses

The two dimensions have been interpreted as the early energy decay, a good measure for which is Early Decay Time (EDT), which specifically considers the energy leak from 0 to -10 dB. This time interval is very short in outdoor spaces, while the audible reverberation can be surprisingly long. Dimension 1 and Early Decay Time are highly correlated (R=0.94). Dimension 2 was found to be related to timbral effects. The integrated autocorrelogram was found to be the closest indication of such spectral periodicities, but only shows medium correlation (R=0.52) with dimension 2.

4.3 Discussion

The present research shows that the Reverberation Time (measuring the decay of energy from 0 to -60 dB), which has been considered as the most prominent aspect in many psychoacoustic studies on reverberation, may actually be much weaker at describing the human perception of acoustic spaces than previously assumed. Indoors, the decay of energy is exponential, which means that Early Decay Time and Reverberation Time are correlated. Outdoors, however, the initial energy leaks much more quickly, which is followed by a exponential tail. The overall time in which the reverberation is audible seems to be much less important than the first drop of energy, however.

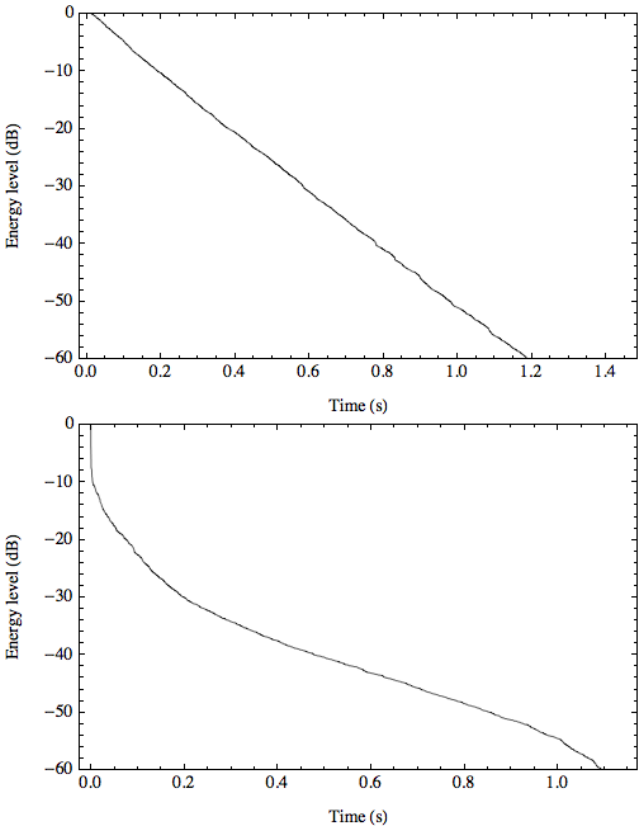


Figure 2: The decay of energy in an indoor and an outdoor space, respectively.

The timbral aspect of reverberation is still hard to assess computationally. Colouration and roughness due to interference effects are perceptually obvious, but are very hard to capture in a measure [8].

One aspect which was neglected in this study was the binaural effect of reverberation. This focus on monaural effects of reverberation means that perceptions of spaciousness or source width cannot be judged on the basis of the data. The binaural effects are certainly important to address in future research, but had to be excluded in this study due to technical reasons.

5. DESIGNING A REVERBERATION INSTRUMENT

As far as reverberation as an audio effect is concerned, it is desirable to have a reverberation model that supports an instrument or voice recording by sustaining energy, and which does not affect the timbre in an unpleasant way.[4]

Re-envisioning reverberation as a musical parameter, however, means that all spaces available to human experience should be accessible: music or sound can travel through various spatial contexts, from a forest, to a basement, to a church. The first step towards this ideal is the implementation of the perceptual map shown in Figure 1 as a parameter space, through which the user can navigate.

Between the techniques of artificial reverberation based on impulse response convolution or modelling, the modelling approach was chosen, since it allows for more flexibility. An improvement on the current delay line modelling techniques is attempted on the basis of the psychoacoustic data.

The energy leak that was found to dominate the perception of reverberating spaces was chosen to be modelled by a leaky integrator:

$$y_t = -Ay_{(t-1)} + x_t$$

The leak coefficient A can be used to vary between a quick drop, or a sustain of energy, and hence model the initial decay development in inside and outside spaces.

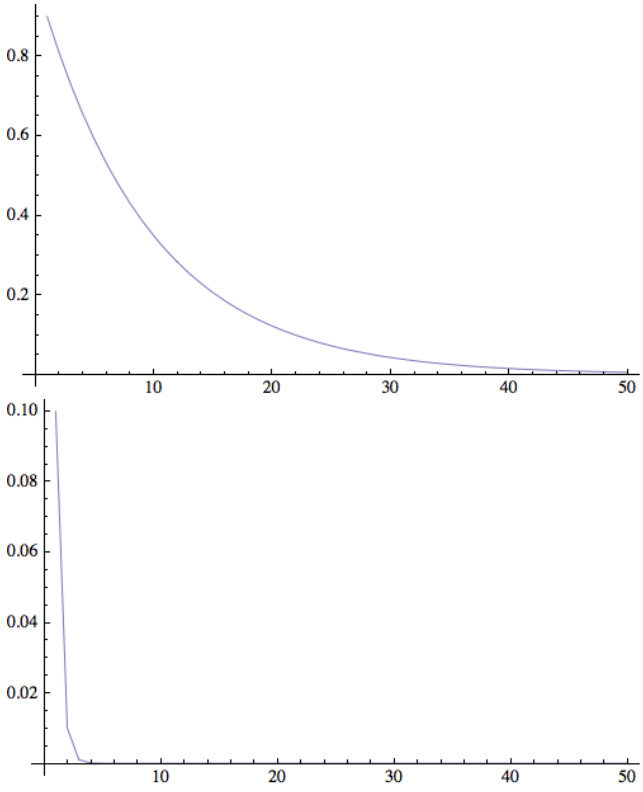


Figure 3: An impulse fed into a leaky integrator with the leak coefficients of 0.9 and 0.1, respectively.

The dimension of timbre is modelled through a set of eight parallel comb filters. The delay time of each comb filter feedback loop can coincide with another, which will lead to a severely coloured spectrum, or the frequency dips of the different filters can be shifted in respect to each other so as to achieve a less coloured signal.

Close attention has to be paid to the filter coefficients, however: it is very difficult to choose them so that the colouration is still perceived as an effect of reverberation. Therefore, the open source Freeverb algorithm[1] was taken as a source for well-tuned filter coefficients.

Through an ensuing series of allpass filters a realistic length of the overall reverberation is ensured. Figure 4 gives a rough overview of the signal chain implemented in the re-

[1] `http://freeverb3.sourceforge.net/`

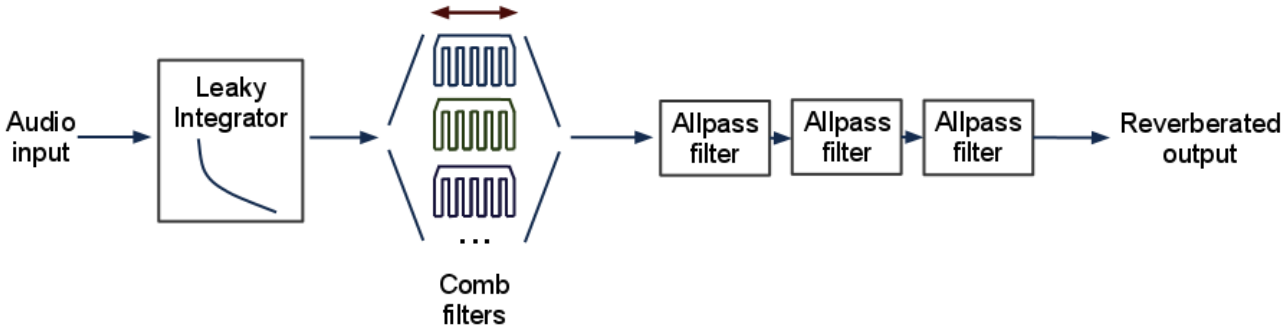


Figure 4: Signal chain of the reverberation instrument.

verberation instrument. The adjustable parameters are the leak coefficient of the leaky integrator, and the shifting of the comb filters, here sketched as combs with prongs representing frequency dips.

While this design rests on existing reverberation algorithms, the control of parameters along the two dimensions of the perceptual map should be a significant improvement to allow intuitive control of reverberation. The reverberation instrument as described here has been realized in SuperCollider and can be controlled with a Wii Motion Plus controller. Other interfaces, for example HID devices or camera tracking systems, are conceivable and can be unproblematically connected via Open Sound Control.

6. CONCLUSION

Controlling reverberation as an instrument is a dream which can only be obtained by understanding how humans perceive and cognitively organize reverberation. It has been shown that current techniques to obtain artificial reverberation do not fulfill this dream, mainly because the physical parameters controlling modelling, or describing an impulse response, are not linked to our perception. Psychoacoustic research can provide this bridge between the physical and perceptual realm, but has sadly mostly been restricted to indoor spaces. The study presented in this paper overcomes this limitation, and provides results which make an instrumental navigation through acoustic spaces more achievable. One important aspect of reverberation, namely binaural effects, have not been considered so far, however. More extended psychoacoustic research is needed, therefore, and its results should be integrated into the model presented here.

7. REFERENCES

[1] Y. Ando. *Architectural acoustics: blending sound sources, sound fields, and listeners*. Springer, 1998.

[2] L. L. Beranek. Concert hall acoustics – 1992. *Journal of the Acoustical Society of America*, 92(1):1–40, 1992.

[3] D. Berkley. Normal listeners in typical rooms. In G. Studebaker and I. Hochberg, editors, *Acoustical Factors Affecting Hearing Aid Performance*, pages 3–24. University Park Press, 1980.

[4] B. Blesser. An interdisciplinary synthesis of reverberation viewpoints. *Journal of the Audio Engineering Society*, 49(10):867–903, 2001.

[5] B. Janssen. Cognitive representation of reverberating rooms and spaces. Master's thesis, Hamburg University, 2008.

[6] P. Lehmann and H. Wilkens. Zusammenhang subjektiver beurteilungen von konzertsälen mit raumakustischen kriterien. *Acustica*, 45:256–268, 1980.

[7] D. G. M. R. Schroeder and K. Siebrasse. Comparative study of european concert halls: correlation of subjective preference with geometric and acoustic parameters. *Journal of the Acoustical Society of America*, 56:1195–1201, 1974.

[8] A. M. Salomons. *Coloration and Binaural Decoloration of Sound due to Reflections*. PhD thesis, Technical University Delft, 1995.

[9] U. Zölzer. *DAFX - Digital Audio Effects*. Wiley, 2002.

Vibrotactile Feedback-Assisted Performance

Lauren Hayes
Department of Music
University of Edinburgh
Alison House
12 Nicolson Square
Edinburgh
EH8 9DF
laurensarahhayes@gmail.com

ABSTRACT

When performing digital music it is important to be able to acquire a comparable level of sensitivity and control to what can be achieved with acoustic instruments. By examining the links between sound and touch, new compositional and performance strategies start to emerge for performers using digital instruments[1]. These involve technological implementations utilizing the haptic[2] information channels, offering insight into how our tacit knowledge of the physical world can be introduced to the digital domain, enforcing the view that sound is a 'species of touch' [14].

This document illustrates reasons why vibrotactile interfaces, which offer physical feedback to the performer, may be viewed as an important approach in addressing the limitations of current physical dynamic systems used to mediate the digital performer's control of various sorts of musical information. It will examine one such method used for performing in two different settings: with piano and live electronics, and laptop alone, where in both cases, feedback is artificially introduced to the performer's hands offering different information about what is occurring musically. The successes of this heuristic research will be assessed, along with a discussion of future directions of experimentation.

Keywords

Vibrotactile feedback, human-computer interfaces, digital composition, real-time performance, augmented instruments.

1. INTRODUCTION

Being arguably the most highly developed of the senses [5], the importance of touch is often, it will here be suggested, erroneously overlooked in human-computer musical systems. This paper examines some possible advantages in exploring the audio-tactile link for practitioners of digital music, and will propose introducing vibrotactile feedback as a new strategy for improving performance in the field. An assessment will be presented of what has been lost, in terms of interaction, in the move from traditional acoustic instruments to commercial interfaces for digital music. This will be followed by a discourse on using vibrotactile signals, directly applied to the performer's hands, as a means of communicating information about the music and score during a performance. The theoretical ideas are put forth in relation to the creative practice of the author; the output of this work being original compositions and improvisations. Links to audio[3]. and video[4] examples of these works have been provided for reference.

[1]Instrument is used here to encompass the entire system which may include: human-computer interface(s), computer, bespoke software, loudspeakers and so on.
[2]Related to the modality of touch.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. SOUND AND TOUCH

2.1 Haptics

When hungarian psychologist Revesz first introduced the word haptic, from the Greek *haptesta* (to touch), in 1931 [3], it was used to describe the process of actively exploring a shape or spatial dimension with the hands, discussed in the context of his research on blindness and its profound effects on the other senses. He contrasted this process with the event of indirectly sensing something on the skin (*ibid.*), such as experiencing differences in temperature or feeling something brush against the body. However, when discussed in terms of human-computer interfaces, the word haptics is often used as an umbrella term encompassing both the active information gathering that Revesz described, as well as the tactile sensations that he classed separately, and additionally, kinaesthetic information about the body in relation to space [12].

Haptic devices are carefully designed interfaces that usually involve some type of actuator or mechanical device, such as small vibrating motors. Their purpose is to improve the translation of gesture between the physical world and the digital realm by considering both the body's kinaesthetic system, which detects position and motor control of muscles and joints, as well as the tactile sensors in the skin, which are extremely sensitive and capable of detecting highly complex patterns of information [6].

2.2 Instruments

2.2.1 Acoustic Instruments

The skin's sensing nerves are most densely collected in the lips and hands [9], [12]; since most acoustic instruments are constructed to be played with the mouth or fingertips, this distribution of sensors in the skin allows for the maximum amount of information exchange. During engagement with the instrument, the performer receives feedback in the form of various resistant forces and vibrations (for example, the

[3]Audio recording of improvisation for laptop using vibro-tactile feedback device: http://soundcloud.com/elleesaich/multifingeredbodyparts
[4]Video of *kontroll* for prepared piano, self-playing snare and live electronics: http://www.vimeo.com/13493035

vibration of guitar strings, along with the force of the fingers on the strings).

This haptic information supports the auditory feedback received through the ears as sound is made. Hence a closed feedback loop is created: the performer makes a sound, which is heard and also perceived physically, judged, and considered before the next sound is made. This process occurs in a very short space of time and is constantly ongoing throughout musical play: the auditory and haptic feedback is immediate, with the latter signals received perhaps almost subconsciously. Certainly while the amount of force used to strike keys while playing the piano is a conscious consideration, vibrations received through the feet may not be consciously perceived, but are no doubt significant in creating the collective feedback information being received. As Cadoz claims, this bidirectional information exchange 'provides us with manipulation possibilities and even signals the nature of the sound phenomenon itself.' [1] Furthermore, this entire process is uniquely private to the performer, compared to noticeable *visual* exchanges that may occur within a group performance setting.

2.2.2 Digital Musical Interfaces

In general, digital interfaces generally offer significantly less feedback to the performer. A MIDI keyboard may feature weighted keys, but cannot reveal any other information about the physicality of the sound being produced, compared to the great resonating body of an acoustic piano.

2.3 Tactile Feedback Principle

In 1978, Claude Cadoz proposed the tactile feedback principle in conjunction with his work at ACROE[5], Grenoble, France, where along with Jean-Loup Florens, he developed the first Retroactive Gestural Transducers (haptic devices). Their aim was to provide new insights into music creation by focusing on the instrument-performer relationship as fundamental to both the learning of the instrument and the development of the music itself, rather than simply providing improved ergonomics of gestural control in sound synthesis [1]. Cadoz claimed that any musical interface into the digital world must succeed on three levels: the gesture used to manipulate the device must be genuine in that the performer must be familiar with the type of movements being used with the controller. Secondly, the device must be able to accurately sense the characteristic behaviour of the gesture and information must not be lost. Finally, he claimed that the device must offer some resistance to the performer, which is in relationship to the nature of the simulated gesture process [1]. He calls this final aspect feedback, and deems it necessary to achieve mastery or perform with finesse.

2.4 Types of Haptic Devices

Human-computer haptic interfaces may be described as any device that incorporates an element of force feedback through actuators: mechanical systems that can offer a wide range of accurate motion, such as motors. Rovan and Hayward distinguish these devices from what they call tactile stimulators [12], which consist of, for example, small groups of pins that tap at the skin, vibrating at controllable frequencies to achieve different intensities. Thus Revesz's distinction between active and passive perception manifests itself in these contrasting systems: the vibro-tactile systems allow the user to passively experience sensations.

There is a huge amount of evidence to suggest that haptic perception can speed up learning [3], [9], thus allowing the relationship between performer and instrument to develop at a much faster rate than without feedback present. When describing his Modular Feedback Keyboard, Cadoz claimed that the aim was to create a 'synthesis of the instrument' [1], as well as the sound. Thus it would allow experimentation, musical play and would successfully couple the performer, instrument and space [1]. As Pedro Rebelo, researcher and composer at *SARC*, Belfast claims, it is useful to view the link between a performer and instrument as a 'multimodal participatory space (and not one of control)' [11]. The following sections will discuss the author's attempts to realise this idea as both composer and performer.

3. FEEDBACK-ASSISTED PERFORMANCE

3.1 Developments

There have been a minimal[6] number of instruments designed with vibrotactile feedback in mind. Marshall and Wanderley, CIRMMT, McGill University, describe the *Viblotar* and the *Vibloslide*[8] which each use small inbuilt speakers to produce both vibration, as well as sound, as feedback for the performer. As with acoustic instruments, in both these examples the sound source is located within the body of the physical instrument itself[7], and not dislocated in loud speakers: the physical feedback emerges concurrently with the sonic.

The aim of using of the vibrations of the speakers was to create "vibrations in a DMI [digital musical instrument] that are produced in a similar way to those of an acoustic instrument". Certainly this approach is an important one, in that it uses the paradigm of traditional instruments as a starting point for introducing haptic information to new digital instruments. The following two case studies offer an alternative approach where the vibrotactile feedback is not intended to emulate the feel of playing acoustic instruments, but rather as a signalling and suggestion system for the performer.

3.2 Case Study: Composition for Prepared Piano and Live Electronics

This work arose out of several compositions for prepared piano and electronics, for solo performer, where the performer would be in control of both the piano and the live electronics. As Emmerson claims, digital music interfaces should be both consistent in their response, as well as sensitive, so that even subtle movements and gestures may be accurately detected and used to affect the sound [4]. Thus it seemed plausible that using a touch-based acoustic instrument, namely the piano itself, as the interface into the digital world could be the solution to achieving mastery of the entire system[8]. By controlling all processes from the piano, the pianist may retain their touch-based sensitivity whilst yielding enough useful control data, via various analyses of the sound, to affect the digital signal. From this emerges what pianist Xenia Pestova describes as a 'further continuation of extended techniques' [10].

3.2.1 Score Following

Building on previous work involving a machine-listening system for prepared piano and live electronics, the goal with the new piece, *kontroll*, was to create a situation where the

[5]Association pour la Création et la Recherche sur les Outils d'Expression

[6]Marshall found instances of vibrotactile feedback implementation in less than 6% of new instruments at NIME from 2001 - 2008[7].

[7]Although external amplification is also permitted to increase sound quality.

[8]Rather than attaching MIDI controllers to the piano, which may disrupt the performance.

Figure 1: Simple glove with vibration motors, which connects to a laptop via an Arduino.

need to look at a laptop screen for visual feedback would be minimized or completely eradicated. In a previous composition, *transient* (2010) it was necessary to watch for the clock, score position and whether various triggers had been activated on the laptop screen. While certain trigger points were flexible in time, they had to occur within a certain time-window, and thus the Max/MSP interface had to been constantly checked. In the subsequent piece, these obstacles would be overcome by sending haptic feedback, in the form of vibrations, to the hands of the performer, providing the required information via a different modality.

3.2.2 Methodology

The score of the composition was created in Max/MSP, where various preset stages were created which would enable or deactivate different DSP modules, and change how control data derived from analysis of the piano's acoustic signal was used. Advancing to a new section would, in most cases, be triggered by the pianist (either performing a particular gesture at a specified dynamic, or by maintaining a specified amount of silence). Other events would advance according to a fixed timeline. Using an Arduino[9] and three small pager motors attached to the left hand via a simple glove, vibrations were sent to the performer indicating:

- a five second warning for an approaching change in score position, increasing in vibration intensity
- a strong short vibration when the performer had successfully triggered a new section of the piece
- the guide tempo of a section.

The pager motors used were Samsung disk coin-type vibration motors[10]. By selecting extremely light motors (0.99 grams each), no additional noticeable weight would be added to the hands of the pianist. The motors were connected directly across the ground and digital/pulse-width modulation pins of the Arduino, as they operate at a meagre 1.5V. Information was sent to the three motors using the Maxuino helper patch[11] for Max/MSP, allowing all computation to be contained inside a single programming environment. Using pulse-width modulation, a very apparent increase in intensity could be experienced.

Motors were fixed onto the glove (which was extremely thin and elasticated), positioned on either side of the back of the hand, with the third positioned directly below on the wrist. This allowed for discreet observable information to be accurately perceived whilst playing.

3.2.3 Outcomes

The result was extremely beneficial to the execution of the performance: the ease with which I could ignore the screen and concentrate on the performance was immediately apparent. The vibration signals were non-evasive and did not distract from the actual playing. There was a strong sense of being fully coupled to the system as a whole, as the score of the piece was being *applied* directly to the body, offering more security in the often unpredictable world of live electronics, and allowing for a more focussed performance.

3.3 Case Study: Improvisation for Laptop and Game Controller

As a trained pianist, most of my musical expressivity involves working with the hands, and thus for laptop performance I often repurpose generic game controllers as my interface. For the second example, the vibrotactile feedback system that was developed for *kontroll* was used in a more active manner, worn in conjunction with a game-pad for laptop improvisation. While used as a signaller of structure in the previous work, the haptic information was now used to direct the performer with more musical, and particularly rhythmic information.

While many game-pads or rumble-pads do offer resistant force-feedback, this was not present in the one that was used. Instead, it was sought to achieve a high level of control using only micro-movements of the hands and fingers. Thumbs pressed on the two joysticks could freeze and loop the sound, but the slightest movement could throw this off.

3.3.1 Tactile Score

The device alone *without* haptics worked fairly well as a solo improvisational tool, triggering samples within Max/MSP which were sliced into segments of several milliseconds, and then processed in various ways. Yet, to create a more interesting deployment of the gestural rhythmic aspect, the vibrotactile glove provided short pulses of 300 milliseconds to the performer indicating that short sounds should be played. The interval between these bursts was determined algorithmically, and changed over time. Thus the illusion of different paces throughout the improvisation was created, along with more unpredictable intervals between gestures. Similarly, longer signals, which would increase with intensity, were sent to indicate that a section should be repeated for the duration of the physical sensation. A variable timeline was established along which either of these two situations could occur, but this would not be known to the performer prior to the start of the piece.

3.3.2 Development

The next part of this work will be to develop musical suggestion which is dependent, at least in part, on what has been, or is currently being played by the performer. Rhythmic patterns would certainly be an obvious starting point here, as these are perhaps the most easily repeatable events when working with unpredictable digital musical instruments. Furthermore, rhythms can be easily represented by short bursts of vibrations. The problem with translating more complex variables, such as spectral content, is not only

[9] An open-source electronics prototyping platform board: www.arduino.cc
[10] Available from www.pagermotors.com
[11] http://www.maxuino.org/

Figure 2: Vibrotactile feedback used in conjunction with generic game controller.

the issue of how to most meaningfully map parameters, but also where to draw the line between useful information and sensory overload.

4. CONCLUSIONS

It is clear that there is a strong case for utilizing the different aspects of the modality of touch within digital music practice in order to challenge ideas about musical creativity, as well as to address the limitations of current systems used to mediate between the digital performer's gesture and sound synthesis. With careful experimentation and clever coupling of instrument and performer, the possibilities for new musical expression are certainly promising. From the examples shown above, it is clear that using vibrotactile feedback for performance strategies is a largely untapped area that is worth proper exploration.

4.1 Future Developments

Further research in this area will examine different parameters that may be successfully used within this type of feedback system, including:

- testing on different parts of the body
- exchanging information amongst a number of performers in an improvisational setting
- mapping other musical parameters to the feedback.

This last topic possibly deserves the most dedication, and work is in progress to develop ways of representing a more complete musical picture tangibly, looking at aspects such as density and spectral shifts[12], to assist with musical interpretation. Indeed Schroeder et al. describe experiments designed for group interaction, where these parameters are represented visually as an abstract image[13]. Moreover, Chang and O'Sullivan suggest looking toward audio-visual theories, such as those proposed by Michel Chion, to develop ways of linking both the tactile and auditory sensations; techniques such as masking and synchronization[13] are proposed [2].

[12]These are perhaps less consciously perceived, compared to amplitude, frequency etc.

[13]For example, synchronization would involve the sound and the sensation occurring at the same time.

4.2 Concerning the Listener

It is hoped that this type of vibrotactile interface can be used with non-performers, who will listen to music, whilst also experiencing it in the form of vibrations. Indeed Gunther and O'Modhrain, implementing this idea with their *Cutaneous Grooves* project, go so far as to suggest that this is a 'potential new art form' [6].

After developing the tactile feedback system and experiencing the ease with which signals can be transferred to the skin, it is hoped to explore the idea of using this information to enhance the listening experience, by coupling it with various physical sensations.

5. REFERENCES

[1] C. Cadoz, L. Lisowski, and J.-L. Florens. A Modular Feedback Keyboard Design. In *International Computer Music Conference*, Glasgow, 1990.

[2] A. Chang and C. O'Sullivan. An Audio-Haptic Aesthetic Framework Influenced by Visual Theory. In T. I. Workshop, editor, *Haptic and Audio Interaction Design*, Jyväskylä, Finland, September 2008.

[3] P. W. Davidson. Haptic Perception. In S. of Pediatric Psychology, editor, *Journal of Pediatric Psychology*, volume 1(3), pages 21–25, 1976.

[4] S. Emmerson. 'Losing Touch?': The Human Performer and Electronics. In S. Emmerson, editor, *Music, Electronic Media and Culture*, pages 194–216. Ashgate, Aldershot, 2000.

[5] M. Grunwald. *Human Haptic Perception: Basics and Applications*. Birkhäuser, Basel, 2008.

[6] E. Gunther and S. O'Modhrain. Cutaneous Grooves: Composing for the Sense of Touch. In *Journal of New Music Research*, volume 32(4), pages 369–381. Swets and Zietlinger, 2003.

[7] M. T. Marshall. *Physical Interface Design for Digital Musical Instruments*. PhD thesis, McGill University, 2008.

[8] M. T. Marshall and M. M. Wanderley. Vibrotactile Feedback in Digital Musical Instruments. In *Proceedings of the 2006 conference on New Interfaces for Musical Expression (NIME)*, 2006.

[9] S. O'Modhrain. *Playing by Feel: Incorporating Haptic Feedback into Computer-Based Musical Instruments*. PhD thesis, Stanford University, CA, 2001.

[10] X. Pestova. *Models of Interaction in Works for Piano and Live Electronics*. PhD thesis, McGill University, 2008.

[11] P. Rebelo. Haptic Sensation and Instrumental Transgression. In *Contemporary Music Review*, volume 25(1/2), pages 27–35. Routledge, February/April 2006.

[12] J. Rovan and V. Hayward. Typology of Tactile Sounds and Their Synthesis in Gesture-Driven Computer Music Performance. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 297–320. Editions IRCAM, Paris, 2000.

[13] F. Schroeder, A. B. Renaud, P. Rebelo, and F. Gualda. Addressing the network: Performative strategies for playing apart. In *International Computer Music Conference*, pages 113–140, 2007.

[14] S. Waters. Performance Ecosystems: Ecological Approaches to Musical Interaction. In E. M. S. Network., editor, *EMS-07 Proceedings*, Leicester: De Montfort, 2007.

Improving User-Interface of Interactive EC for Composition-Aid by means of Shopping Basket Procedure

Daichi Ando
Tokyo Metropolitan University
6-6, Asahigaoka, Hino
Tokyo, Japan
dandou@sd.tmu.ac.jp

ABSTRACT

The use of Interactive Evolutionary Computation(IEC) is suitable to the development of art-creation aid system for beginners. This is because of important features of IEC, like the ability of optimizing with ambiguous evaluation measures, and not requiring special knowledge about art-creation. With the popularity of Consumer Generated Media, many beginners in term of art-creation are interested in creating their own original art works. Thus developing of useful IEC system for musical creation is an urgent task. However, user-assist functions for IEC proposed in past works decrease the possibility of getting good unexpected results, which is an important feature of art-creation with IEC. In this paper, The author proposes a new IEC evaluation process named "Shopping Basket" procedure IEC. In the procedure, an user-assist function called Similarity-Based Reasoning allows for natural evaluation by the user. The function reduces user's burden without reducing the possibility of unexpected results. The author performs an experiment where subjects use the new interface to validate it. As a result of the experiment, the author concludes the the new interface is better to motivate users to compose with IEC system than the old interface.

Keywords

Interactive Evolutionary Computation, User-Interface, Composition Aid

1. INTRODUCTION

1.1 Composition-Aid IEC

In Interactive EC for art-creation by Dawkins[4], both advantages of the stochastic techniques have been consistent with the deterministic advantage.

In an IEC method, an initial population is generated randomly according to the user's instructions. Then the population converge based on the interactions between the user and the System. Eventually, the user get results that don't need to be corrected anymore. Also, IEC systems apply genetic operators, such as crossover and mutation, on a random basis, to allow the user to discover unexpected and potentially promising results from the stochastic methods.

Many gene representations and user-interfaces have been tried for application of IEC to composition assistance. A general review on the application of EC, especially GA and GP, to composition can be found in [2]. Also there are important studies about musical structural chromosome for composition[3, 5, 6, 7].

1.2 Problems of User-Interface of Composition-Aid IEC

1.2.1 Identifying individuals

In spite of their usefulness face poorly defined fitness functions, composition-aid systems from previous works have some issues regarding the user interface. The interface of most of these systems is constructed using only CUI. This is because the user interface for a basic IEC composition-aid system requires only two elements: one output, the piece playback, and one input, the fitness value. Also, most systems are built with only their own developer in mind, who usually is a composer who completely understands the gene representation and the composition process of their own system. In many cases, the system is designed only for their own creations. Thus expensive user-interfaces are not required for them.

When the system has a GUI interface, it consists only of a playback button (sometimes also a stop button) and a radio button to input the fitness value.

Consequently, there is a small number of works on IEC composition-aid systems. Among the most significant works the CoNGA[7] system uses a Multi-Field user-interface to combine small rhythms patterns and functions that connect these patterns.

The most important point of composition-aid IEC is that the object being generated is a time-based media. The key difference between the composition-aid IEC and IEC for normal art creation, such as computer graphics, is that the characteristics of time-based media cannot be properly displayed in a 2D user-interface.

The first means that to evaluate time-based media it is necessary to listen to the whole piece at least one time, if there is only a playback button on the IEC interface. In this way, it is difficult to identify different individuals at a glance. Therefore in music IEC, the user should listen all individuals carefully from begin to end, and so serious user's burden emerge.

By contrast, visual art can be displayed in such a way that the user can visualize all the individuals at the same time, and decide their fitness values at a glance.

1.2.2 Problems of Applying GA Scenario

Figure 1 shows a typical procedure named Genetic Algorithm Scenario to evaluate individuals on almost composition-aid IEC. The our developed interface is applied on the such like composition-aid IEC system.

To evaluate all individuals with the existing GUI for composition-aid IEC , the user repeats (1) process in the figure. The (1)

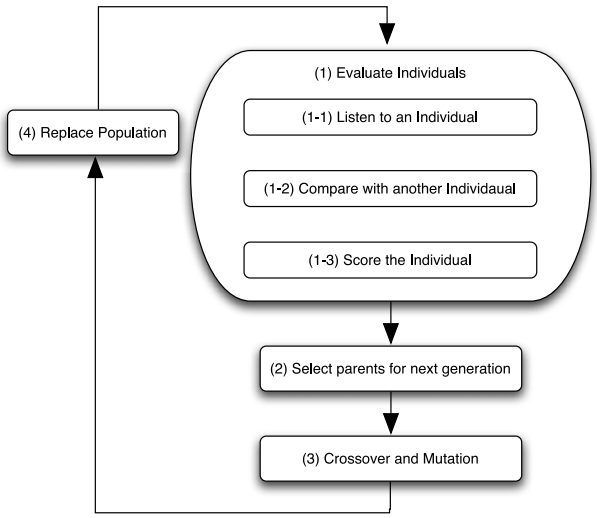


Figure 1: Procedure of traditional GA Scenario

process consists of not only 2 process, (1-1) Listen to an individual and (1-3) Score the individual. Almost the user compares the individual with another individuals to give score. The comparison process corresponds to the (1-2) process in the figure. Also these three processes are executed not always by turn. Sometime, an evaluated individual given score again after comparison with an another individual. The repeat processes in no particular order without serious burden is important for composition-aid IEC interface.

2. SHOPPING BASKET FOR COMPOSITION-AID IEC

2.1 Shopping Basket Procedure

To solve these problems of past systems, the author proposed a new interface "Shopping Basket" for musical composition-aid IEC. Shopping Basket improves procedure of evaluating individual, also the interface does not reduce unexpected good artistic results.

Features of the proposed interface are as follows:

1. Divided evaluating procedure based on Shopping Style to reduce evaluation, listening, burden.
2. Evaluating Individual with drag & drop area change.
3. Similarity based reasoning (SBR) by means of moving similar individuals into other areas at the same time. The SBR does not prevent that unexpected good artistic results emerge.

Figure 2 shows an overview of the Shopping Basket procedure IEC interface. Individuals are displayed as colored sphere icons. The user can listen individual, clicking the individual icons. Also the Shopping Basket interface is divided into five area. To evaluate individuals, the user should move individual icon into other areas by drag and drop.

The Shopping Basket procedure is based on moving state of goods in shopping basket. Figure 3 shows the proposed procedure, individual icon moving between areas is shown in the figure 2.

1. Listening to individuals lightly in "Un-evaluated Area" (1), moving un-favorite individuals to "Dust Box"(5) and favorite individuals to "Compare Area"(2).
2. Listening to individuals carefully in "Compare Area".

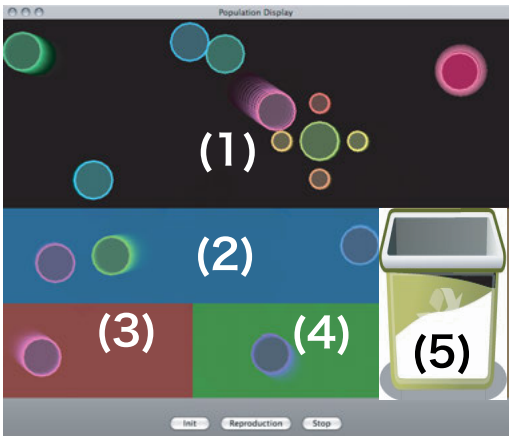


Figure 2: Overview of Shopping Basket IEC Interface

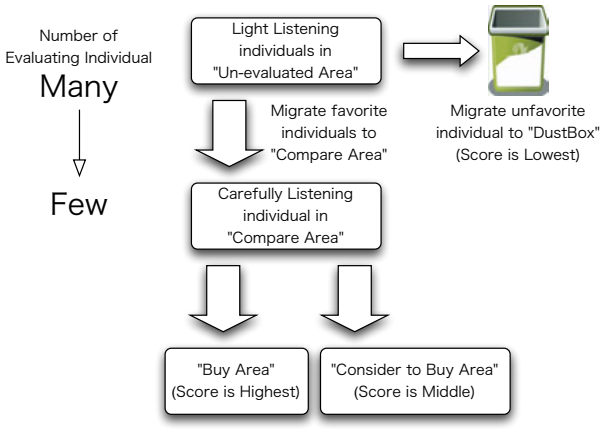


Figure 3: Evaluating Procedure of Shopping Basket

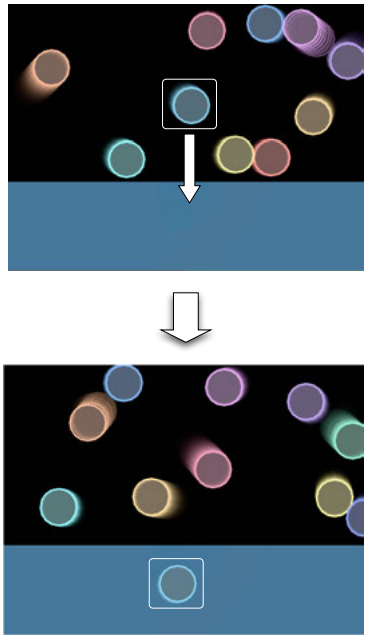


Figure 4: Evaluation - changing area in Shopping Basket

3. Move favorite individuals to "Buy Area"(3).
4. Move not so favorite individuals to "Consider to Buy Area"(4).
5. Indicate system to reproduction.

Score, fitness, which given individuals in "Buy Area" is highest, in "Consider to Buy Area" is middle. Individuals in "Dust Box" are given lowest score.

Figure 4 shows area change movement of individual icon to evaluate. In this case, the user drags an individual icons in white frame at the upper figure, then drops the icon into bottom area in the lower figure.

Most difference between the proposed the Shopping Basket evaluating procedure and the traditional GA Scenario procedure that shown as figure 1 is listening burden of the user. In the GA Scenario procedure, all individuals should be listened very carefully any number of times. However, in the proposed procedure, un-favorite individuals are removed in the first process of the procedure without carefully listening. Thus the user can pay attention to only few their favorite individuals.

Also in the GA Scenario procedure, as mentioned before in section 1.2.2, the user should re-decide fitness score over and over, (1) process in figure 1 repeated by the user. This is due to that fitness score is relative evaluation in a population; therefore measure of fitness score is fluctuating sharply evaluating in a generation in the GA Scenario. On the other hand, in the Shopping Basket, evaluation procedure is not repeated. The user marks fitness on individuals as absolute scale naturally. The author expects that this difference that the actual number of times the user listening individuals is useful for reducing user's burden. It is notable that the number of times to evaluate is treated as a measure for efficiency tests of non-interactive EC algorithms.

In addition, there is another difference is that the area change evaluation. In the traditional GA Scenario, the user should give fitness score to every individual. The author expects that the evaluation by means of area change of individual icon that reduces user's mental burden. This is due to some our preliminary experiments results about IEC suggests such possibility.

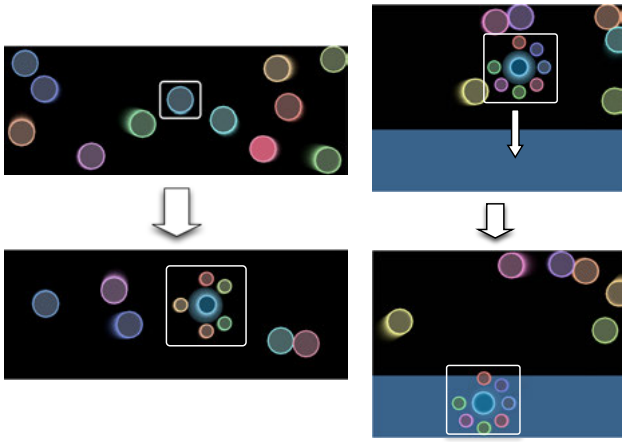


Figure 5: SBR : Function to draw similar individuals up to the target individual, then change area simultaneously with similar individuals.

2.2 Similarity-Based Reasoning

Figure 5 shows work flow of Similarity-based Reasoning (SBR) function. The user can draw similar individuals up to the target individual. In the left-up figure, a target individual is displayed walled in white frame, then similar individuals in same area are drawn up to the target individual as shown in the left-down figure. Then, the user can move all individual that drawn up into other area simultaneously as shown in the right figure.

It means that the user can give the same fitness to similar individuals at once. This function works as SBR. Also the user can listen the drawn up individuals at any time by mouse over action. This means that the user does not failed to catch unexpected good results.

3. VALIDATION OF SHOPPING BASKET

3.1 Experimental Detail

The author performed an experiment where subjects use the proposed interface to validate that the proposed interface and evaluation procedure, the area change of individual icon, the shopping basket procedure and SBR provides functions which reduce user's burden by means of a subjective questionnaire testing.

As composition-aid EC engine, "CACIE" system[1] was used. CACIE is a one of IEC composition-aid system by means of Interactive Genetic Programming.

To make comparative study, two exists user-interfaces are used.

The first one is "Ordinary" type interface which applies normal IEC into the music creation, shown as figure 6. Only two functions, "Play" button to listen individual and slider to which give fitness, are provided by the interface.

The second one is "Circle" type interface, shown as figure 7, which individuals are displayed as sphere icon too. Also playback function is provided as clicking individual icon the same as the Shopping Basket. However, there is differences between the Shopping Basket and the Circle type interface about how to give the fitness. In Circle type interface, the position of an individual icon, its distance from the center of the circle, determines the fitness value of that individual. A higher fitness degree is indicated by a position nearer the

Figure 6: "Ordinary" type interface.

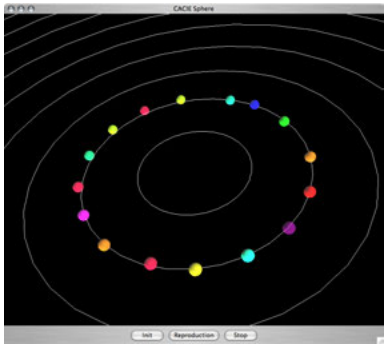


Figure 7: "Circle" type interface.

center of the circle, and a lower fitness value is indicated by moving the individual away from the center. The initial position indicates a neutral fitness value. It means that clear difference between the Shopping Basket and the Circle type is that existing of divided areas.

The number of subject is ten. Each subject had tests the three interfaces in random order. Each after of test for one interface, subject filled out the questionnaire.

3.2 Subjective Questionnaire

Each question of the questionnaire is as follows:

1. It was difficult to distinguish each individual.
2. I listened individuals without stress.
3. I could identify each individual easily, it was easy to listen to compare individuals.
4. Action of evaluation was easy to understand.
5. I was lost in thought to evaluate for a long time.
6. I enjoyed this composition time.
7. I hope to use this interface to composition.

Each subject answered these questions in 5 degree, 1.Strongly Agree, 2.Agree, 3.Cannot judge, 4.Disagree, 5.Strongly Disagree.

Figure 8 shows result of questionnaire that average of score of all subjects. Also results of applying ANOVA(5%) to the questionnaire answers are shown in table 1, three significant differences between the proposed Shopping Basket and the others are occurred. Total ANOVA result shows that the Shopping Basket surpass other interfaces.

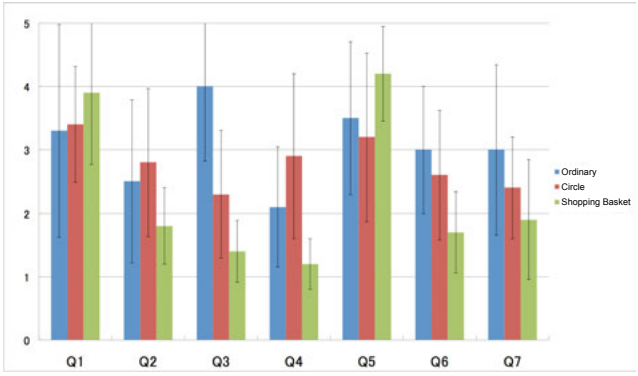


Figure 8: Result of questionnaire. Average of each questions are displayed. Error bar means standard deviation.

Q3	Shopping Basket = Circle < Ordinary
Q4	Shopping Basket < Circle = Ordinary
Q6	Shopping Basket < Circle = Ordinary

Table 1: Result of ANOVA(5%). These 3 questions have significant differences.

4. CONCLUSION

In this paper, the author has presented improvements interface of composition-aid IEC by means of Shopping Basket procedure. As the result of subjective evaluation experiment, the proposed Shopping Basket reduces user's burden was confirmed.

The theme that reducing user's burden without filtering unexpected results have been critical theme of studies about composition-aid IEC. We need to continue to study about user-interface to realize composition-aid IEC for actual use.

5. REFERENCES

[1] D. Ando, P. Dahlsted, G. Nordahl, and H. Iba. Computer aided composition by means of interactive gp. In *Proceedings of International Computer Music Conference 2006, New Orleans, USA*, pages 254–257. ICMA, October 2006.

[2] A. R. Burton and T. Vladimirova. Generation of musical sequences with genetic techniques. *Computer Music Journal*, 24(4):59–73, 1999.

[3] P. Dahlstedt and M. G. Nordahl. Augumented creativity: Evolution of musical score material, 2004.

[4] R. Dawkins. The evolution of evolvability. In C. G. Langton, editor, *Artificial Life*, pages 201–220. Addison-Wesley, 1989.

[5] P. Laine and M. Kuuskankare. Genetic algorithms in musical style oriented generation. In *Proceedings of First IEEE Conference on Evolutionary Computation*, pages 858–861, Washington D.C., 1994. IEEE.

[6] J. B. Putnam. Genetic programming of music. Unpublished manuscript. Socorro, NM: New Mexico Institute of Mining and Technology, 1994.

[7] N. Tokui and H. Iba. Music composition with interactive evolutionary computation. In *Proceedings of 3rd International Conference on Generative Art (GA2000)*, 2000.

BioRhythm: a Biologically-inspired Audio-Visual Installation

Ryan McGee
PhD Student
Media Arts and Technology
University of California,
Santa Barbara
ryan@mat.ucsb.edu

Yuan-Yi Fan
PhD Student
Media Arts and Technology
University of California,
Santa Barbara
dannyfan@mat.ucsb.edu

Reza Ali
PhD Student
Media Arts and Technology
University of California,
Santa Barbara
syedali@mat.ucsb.edu

ABSTRACT

BioRhythm is an interactive bio-feedback installation controlled by the cardiovascular system. Data from a photoplethysmograph (PPG) sensor controls sonification and visualization parameters in real-time. Biological signals are obtained using the techniques of Resonance Theory in Hemodynamics and mapped to audiovisual cues via the Five Element Philosophy. The result is a new media interface utilizing sound synthesis and spatialization with advanced graphics rendering. BioRhythm serves as an artistic exploration of the harmonic spectra of pulse waves.

Keywords

bio-feedback, bio-sensing, sonification, spatial audio, spatialization, FM synthesis, Open Sound Control, visualization, parallel computing

1. INTRODUCTION

Hemodynamics is the study of blood flow and circulation. Resonance Theory in Hemodynamics (RTH) [12, 7] provides scientific evidence of the relationship between harmonic peaks of blood volume change signals and visceral organs. The spectra and frequency selectivity of arterial beds in organs were found to change profiles following specific patterns with ligations of different arteries [7]. Three primary concepts summarize RTH. First is the measurement of a subject's physiological condition by palpation sensors and harmonic analysis of the resulting pulse waveform using objective signal processing techniques. Second is the perspective that there exists a direct relationship between the efficiency of the cardiovascular system and the development of meridians within a species. Biologically, meridians are pathways for the flow of qi (pronounced "chi"), the Chinese term for psychophysical energy. Finally, animal and clinical studies show the specific relations between visceral organs and pulse harmonics [7].

Five Element Philosophy (FEP) [11] provides a mapping from visceral organs to musical pitch, color, and cardinal direction. Table 1 combines Resonance Theory in Hemodynamics with FEP to create the mappings used in BioRhythm. We use FEP only as a metaphor for creating the artistic vocabulary and bio-audio-visual mappings within this work. BioRhythm is the first known attempt to establish a relationship between RTH and FEP.

Table 1: Summary of RTH and FEP

Harmonic	0	1st	2nd	3rd	4th
Organ	Heart	Liver	Kidneys	Spleen	Lungs
Color	Red	Green	Black	Yellow	White
Pitch	G	E	A	C	D
Direction	South	East	North	Center	West

1.1 Background

While the sonification of biological data for artistic purposes has a long history, the majority of projects have focused on electroencephalographic (EEG), electrocardiographic (EKG), electromyographic (EMG), or some combination of several sensors [10, 9]. In addition to aesthetic exploration, other studies have focused on biological sonification as a tool for diagnosis [5]. With BioRhythm the goal was to extract as many audio-visual mapping parameters as possible using a single fingertip photoplethysmograph (PPG) sensor. The unobtrusive PPG interface is also more conducive to audience interaction than other sensors.

1.2 Biological Sensor

A photoplethysmograph (PPG) is an optical sensor that measures blood volume changes by illuminating the skin with an LED and detecting the amount of light transmitted or reflected to a photodiode. The PPG used in BioRhythm takes measurements on the index finger and is used as the primary source to drive audiovisual generation. The spectrum of a PPG signal is characteristic of the harmonic spectra used in RTH. In addition, the optical PPG has the advantage of being easy to use for public installations.

Figure 1: PPG Sensor (BIOPAC Systems, Inc.) [1]

1.3 Harmonic Analysis of the PPG Waveform

The harmonic modulus is defined as ratio of amplitude of the harmonic to that of the fundamental frequency. BioRhythm uses RTH and FEP to map the harmonic moduli of a PPG signal (Figure 2) into five frequencies, colors, and directions that serve as the basis for sonification and visualization techniques.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

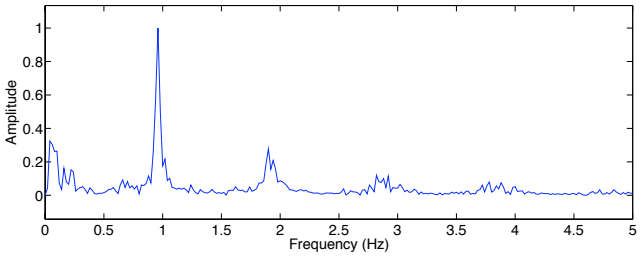


Figure 2: Spectrum of typical PPG Signal

1.4 System Integration

Real-time physiological signal acquisition to sonification and visualization is accomplished with parallel computing between three laptops via Open Sound Control (OSC) [13]. One laptop captures and processes the raw PPG data using BIOPAC software [1] then uses Max/MSP to send the raw data along with spectral information via OSC. Two other laptops receive OSC data and generate the sonification and visualization respectively.

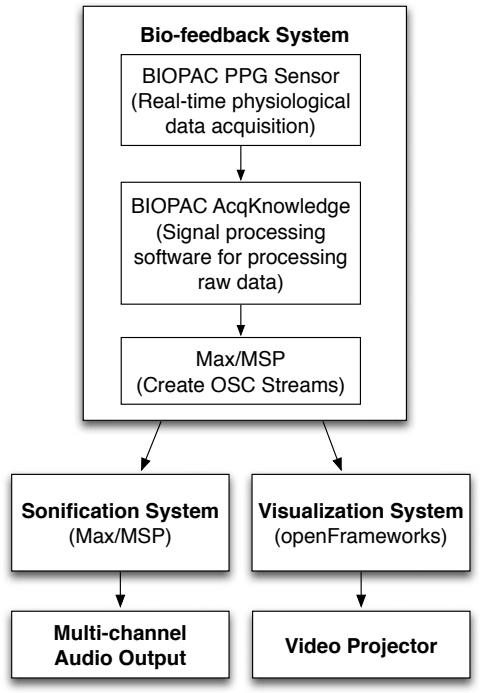


Figure 3: System Integration

2. SONIFICATION

The sonification for BioRhythm is composed of 3 separate sonic layers using an intricate combination of FM synthesis, spatialization, filtering, and delay lines implemented in Max/MSP. Parameters received via OSC are the raw PPG signal, heart rate (in beats per minute), interbeat interval (time between heart beats in milliseconds), and the amplitudes (0-1) of the five PPG harmonics. The sonification software also computes each user's average heart rate, thus, there is a brief learning phase for before the sonification begins for each new user. Figure 4 provides and overview of the sonification in BioRhythm.

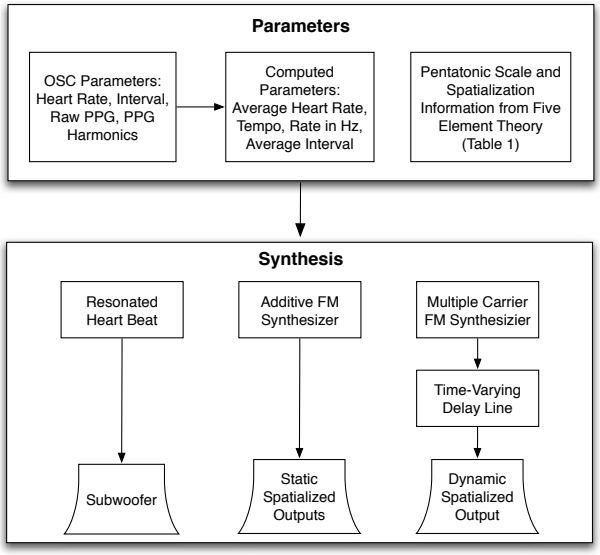


Figure 4: Sonification in BioRhythm

2.1 Heartbeat Layer

The first layer takes an iconic approach to sonification [6] by representing the user's heartbeat with a sound sample of a heartbeat triggered at each peak of the PPG signal. The sample is equalized with frequency peaks in the 40-80hz range corresponding to G, E, C, D, and A musical notes as defined by FEP. There is no sound synthesis or spatialization at this layer, but the heartbeat layer provides the user with a simple, clear biofeedback response while the other two layers contain more depth.

2.2 FM Synthesis

The second and third layers of sound use a unique approach to sonification that combines model-based and parameter-based methods [6]. If we think of the sonification software as an instrument, then the incoming data not only plays the instrument, but also defines and reshapes the instrument. This is accomplished through the use of frequency modulation (FM) sound synthesis. The basic elements of FM synthesis are a carrier frequency (the fundamental frequency), a modulation frequency (the rate at which the carrier frequency will vary), and the modulation index (the amount of frequency deviation from the carrier which directly corresponds to the number of resulting partials). When the ratio of the carrier to modulator frequency is an integer a harmonic sound results. For non-integer ratios the sound is inharmonic.

2.3 Additive FM Layer

The foundation of the second sonic layer sums five simple FM synthesizers, a technique known as additive FM synthesis. Each of the carrier oscillators is set to a frequency corresponding to one of the five notes of the FEP pentatonic scale specified in Table 1. The modulator frequencies begin at a 1:1 ratio with the carrier. The modulators continuously adjust themselves so that the ratio of the average heart rate to current heart rate matches the carrier to modulator ratio. Thus, the modulator frequency changes with each heartbeat. Due to the properties of FM synthesis, the sounds become more harmonic as the user's heartbeat closely matches their computed average. Large changes from the average heart rate will result in more inharmonic sounds. The amplitudes of the 5 harmonic filters are scaled

to a modulation index between 0 and 25 for their respective modulators. Thus, strong presence of a particular harmonic in the PPG signal produces a high modulation index that results in a harsher sound with more high frequency components. The output of each FM synthesizer is given a static position corresponding to a cardinal direction specified in Table 1.

2.4 Multiple Carrier FM Layer

The third layer of sound implements multiple carrier FM synthesis (MCFM), multiple carriers sharing a single modulator). Again, each of the carrier frequencies corresponds to a note in the FEP pentatonic scale. However, in this layer the user's current heart rate is converted to Hertz and used as the single modulation frequency. The raw PPG signal is scaled to an index between 0 and 25 for the modulator. Higher amplitudes of raw PPG data indicate higher blood pressure for the user and will raise the modulation index of the sound, resulting in more high frequency components. Likewise, users with lower blood pressure will experience a calmer sound with less high frequency components.

2.5 Spatialization and Movement

Further interpretation of the third layer lies in the timing and spatialization of the synthesized sounds. The output of the MCFM synthesizer connects to a delay line with a variable delay time set by the interbeat interval. Thus, slower heart rates produce longer delays that echo the dry sound. If the heart rate suddenly increases it will be echoed by several short time-delayed versions of the sound. The time varying delay lines shift the frequency of the sound, which produces an effect similar to Doppler shift. This delay line implementation makes apparent subtle changes in heart rate that cannot be interpreted by simply listening to the heartbeat of the first layer. Lastly, this layer is highly spatialized as a single point source moving rapidly around the user. A direction (azimuth) and distance is computed from the weighted average of harmonic levels that correspond to cardinal directions. The panning algorithm used is original but similar to distance-based amplitude panning (DBAP) techniques [8].

2.6 Sonification Summary

To summarize the sonification, we have a simple heart beat sound providing the underlying rhythmic interpretation and deep bass frequencies, a second layer that represents change through tonal timbre and chord formation, and a third fluttering treble layer that focuses more on time delays and dynamic spatialization to represent change. Due to the extreme sensitivity of the PPG, a static sound cannot be achieved even in the user's most restful state. Thus, users are not required to elevate their heart rate or dramatically change their physiological state to hear interesting results (though many have fun doing so). Each individual produces their own unique choreography of sounds due to parameters of their unique heart beat or biorhythm.

3. VISUALIZATION

The visualization consists of a single abstract organic form produced by algorithmic methods focused on distorting of a perfect sphere. As with the sonification, the incoming bio-signals are mapped according to RTH and FEP, which leads to an extremely coherent synchronization between auditory and visual cues in the installation. The visualization receives the same parameters as the sonification via OSC and runs as a C++ OpenFrameworks [4] application.

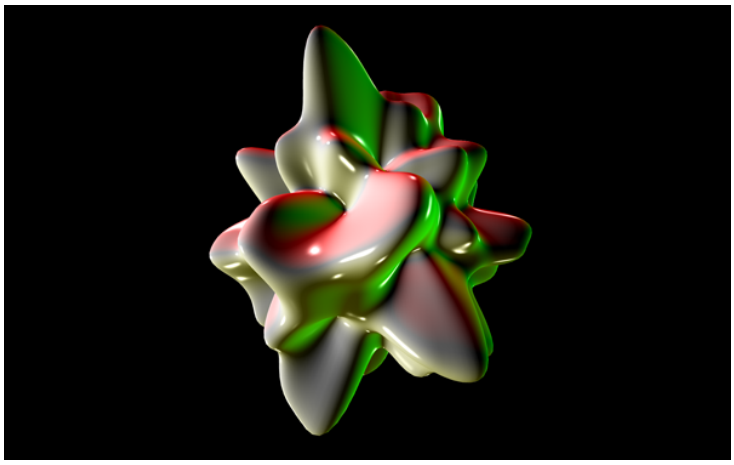


Figure 5: BioRhythm Visualizaton

3.1 Aesthetic

The development of an organic aesthetic was pursued because of the biological nature of the research. This organic form steers away from traditional data graphs and plots, so to engage the installation's audience in a deeper and more anthropomorphic manner. The blob-like extrusions in the visual aesthetic help to emphasize the human body, its organs and their fluctuations, as described in RTH.

3.2 Mapping

The base radius of the sphere reflects the amplitude of the raw PPG data and thus the overall form appears to "thump" in unison with the user's heard beat and sonification. A 3D Perlin noise function distorts the vertices of a sphere as the user's heart rate departs from equilibrium and the sonification produces more inharmonic sounds. If the user can steady their heart rate, then they will be able to replicate a pseudo-perfect sphere along with a harmonic sonification. Otherwise, as with the sound, the form will fluctuate and warp according to the blood pressure in the user's finger.

Four lights add color to the visual scene. Green, red, black (absence of light), yellow, and white lights are placed according to the cardinal directions given by FEP. The presence of any color varies with the amplitude of its harmonic along with the presence of a FM carrier frequency in the sonification.

4. PUBLIC INSTALLATIONS

BioRhythm has been publicly exhibited at MindShare Los Angeles [3] and at the Media Arts and Technology End of Year Show at the University of California, Santa Barbara (UCSB) [2]. Both events regularly draw hundreds of artists, engineers, scientists, and others interested in new media at intersection of art and technology.

4.1 Hardware

The original BioRhythm installation at UCSB used a video projector and accompanying array of 32 speakers in an upwards-pointing semi-circle on the floor around the user. A stereo version has also been implemented for venues where larger surround setups are not possible. The fingertip PPG sensor hangs from the ceiling by a thin wire.

4.2 Interactivity and Feedback

The public feedback we receive is generally positive, and users are often comment on how well the sounds match the visuals. Typical behavior after waiting for the 15 second learning phase is that people either attempt to remain

Figure 6: 32 Channel Installation at UCSB

calm to settle the system, attempt to elevate their heart rate through movement to provoke dramatic changes, or attempt some "hack" of the system. For instance, realizing that blood flow is being measured in the finger, some users trigger audio-visual reactions by clenching their finger or fist. In fact, one man attempted to cut off circulation in his girlfriend's hand, and when he released her wrist there was indeed a dramatic change in timbre and spatialization with an explosive warping of the visual form (similar to Figure 7). Other users realize the extreme sensitivity of the PPG sensor and squeeze or rub the sensor to produce extreme effects. Though these hacks are quite interesting, we find that each individual produces interesting time-varying results when remaining still. The sensitivity of the system is such that the user would have to be laying down nearly unconscious to achieve a constantly harmonic sound with spherical shape.

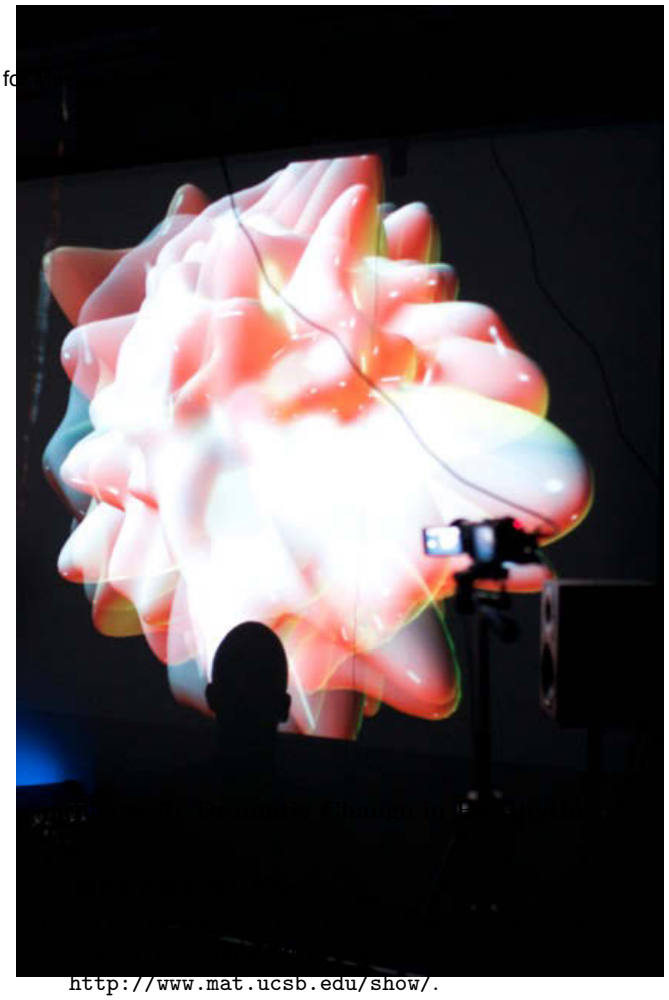
5. CONCLUSIONS

BioRhythm explores the use of a biofeedback sensor for an interactive audio-visual installation. The simple velcro fingertip sensor makes BioRhythm suitable for public installations with hundreds of users. The sonification involves dynamic changes in pitch, timbre, rhythm, and spatialization most notably through the use of FM synthesis, delay lines, and multichannel panning. The visualization projects concurrent movements of shape and color. Users are able to observe audio-visual reactions in the installation corresponding their unique physiological states.

Future work involves improved sensor calibration, searching for patterns within pulse spectra, and the use of alternative sound synthesis methods. RTH and FEP have provided one mapping from biological data to sound and visual domains, but they are only a starting point, and we are excited to explore newly proposed mappings from artists and scientists alike.

6. ACKNOWLEDGMENTS

We are grateful to JoAnn Kuchera-Morin (UCSB) and Alan Macy (BIOPAC Systems, Inc.) for hardware support and to Dan Overholt (Aalborg University) for software support.



http://www.mat.ucsb.edu/show/.
[3] Mindshare Los Angeles. http://www.mindshare.la/.
[4] openFrameworks. http://www.openframeworks.cc/.
[5] M. Ballora, B. Pennycook, P. C. Ivanov, L. Glass, and A. L. Goldberger. Heart Rate Sonification: A New Approach to Medical Diagnosis. *Leonardo*, 37(1):41–46, 2004.
[6] T. Hermann. Taxonomy and Definitions for Sonification and Auditory Display. *Proc. of the 14th ICAD, Paris*, 2008.
[7] W.-K. Lin Wang, Yuh-Ying; Hsu, Tse-Lin; Jan, Ming-Yie; Wang. Review: Theory and Applications of the Harmonic Analysis of Arterial Pressure Pulse Waves. *Journal of Medical and Biological Engineering*, 30(3):125–131, 2010.
[8] T. Lossius, P. Baltazar, and T. de La Hogue. DBAP-Distance-Based Amplitude Panning. In *Proceedings of 2009 International Computer Music Conference, Montreal, Canada*, 2009.
[9] Y. Nagashima. Bio-sensing Systems and Bio-feedback Systems for Interactive Media Arts. *Conference on New Interfaces for Musical Expression*, pages 48–53, 2003.
[10] D. Rosenboom. *Biofeedback and the Arts: Results of Early Experiments*. Aesthetic Research Centre of Canada, 1974.
[11] I. Veith. *Introduction to the Nei Ching (The Yellow Emperor's Classic of Internal Medicine)*. University of California Press, 1966.
[12] Y. Wang, S. Chang, Y. Wu, T. Hsu, and W. Wang. Resonance, The Missing Phenomenon in Hemodynamics. *Circulation Research*, 69(1):246–249, 1991.
[13] M. Wright and A. Freed. Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. In *Proceedings of International Computer Music Conference*, pages 101–104, 1997.

Vibration, Volts and Sonic Art: A practice and theory of electromechanical sound

Jon Pigott
Cardiff School of Art and Design
Bath Spa University
www.sonicmarbles.co.uk
jpigott@uwic.ac.uk

ABSTRACT
This paper explores the creative appropriation of loudspeakers and other electromechanical devices in sonic arts practice. It is proposed that this is an identifiable area of sonic art worthy of its own historical and theoretical account. A case study of an original work by the author titled *Infinite Spring* is presented within a context of works by other practitioners from the 1960's to present day. The notion of the 'prepared speaker' is explored alongside theories of media archeology, cracked media and acoustic ecology.

Keywords
Electromechanical sonic art, kinetic sound art, prepared speakers, *Infinite Spring.*

1. INTRODUCTION
In 2004 I began experimenting with the physical manipulation of loudspeaker cones in my music and creative sound practice. During a residency at WRAP arts centre, Bergen, in 2005 I built an installation consisting of speakers with torn and broken cones, I used aluminium foil to cover speakers and attached metal objects to speaker cones. In my *Visual Speakers* of 2006 I coupled a speaker cone to an acetate sheet such that it would audibly and visibly vibrate, and used another to act as a vibrating switch contact to flash LED lights. In the *Sonic Marble Run* (2007) and *Infinite Spring* (2010) I used beads, foil cups, springs and other objects to mechanically modify electrically transmitted sound. I use the term 'prepared speaker' for these devices in reference to the prepared piano pieces of John Cage and other examples of 'prepared' instruments where objects are mechanically coupled to traditional musical instruments to alter sonic behavior in some way.

This creative appropriation of the loudspeaker reveals a general interest and focus on electromechanical transduction within sound and sonic arts practice, which I aim to explore in my work. By preparing and extending speakers and other electromechanical devices I hope to identify them as sound sources, and raise questions on the relationship between the mechanical nature of the speaker cone and the electromagnetic nature of the energizing signal driving it. Electricity is ubiquitous as an energy source for all manner of human sonic activity yet the acoustic transmission of sound remains a mechanical phenomenon. The meeting, transduction and interplay between these two forces offers rich creative possibilities which I hope to exploit in my work and explore in the work of others.

This paper provides a context for this approach to creative sound practice offering possible ways to navigate the aesthetic and technical concerns of working in this way. A brief survey of works, techniques and technologies that can be said to exist in this arena is provided along with a case study of my recent piece *Infinite Spring* which makes extensive use of both prepared speakers and motors to create an electromechanical sonic environment (see figs 1- 4).

2. CONTEXT
2.1 Brief Survey of Works
Courting the unique electromechanical qualities of the loudspeaker in sonic art and experimental music is nothing new. It was the notion that 'the loudspeaker should have a voice which was unique and not just an instrument of reproduction but as an instrument unto itself' [7] which inspired David Tudor (1926-1996) to make the installation and performance piece *Rainforest* in various incarnations between 1966 and 1974. The piece, still recreated today, uses cone-less loudspeakers as transducers, to enable an electrical signal to mechanically excite objects such as bedsprings, slinkys and sheet metal. The mechanical activity of each of these objects is then amplified using Piezo electric contact microphones [5]. The piece, in all its guises, is documented by Matt Rogalsky who uses the term 'sculptural speakers' to describe the Tudor devices [5].

In *Music For Solo Performer* (1965), Alvin Lucier directly coupled loudspeakers to percussion instruments such as snare drums and gongs such that they would be mechanically excited by the amplified brainwaves of a performer wearing electrodes. Lucier comments that 'the brainwave piece is as much about resonance as it is about brainwaves' ([8], p. 204). 'I'm trying to make the connection between sympathetic vibration, which is a physical thing, and the next idea is the room as a speaker' ([8], p. 205). Lucier is promoting the idea that the speaker preparations in this piece go beyond the percussion instruments and could be extended to include the influence of the room as an acoustic space.

The notion of using space as an extension to the loudspeaker is further explored by Lucier in '*I am sitting in a Room'* (1970). Here he cycles a recording of his own voice through a loudspeaker in a room, recording the result and replaying it back into the same room contiguously until the signal degrades leaving just the sonic resonances of the room / loudspeaker combination exposed. One of Lucier's inspirations for the piece was the testing procedure that loudspeaker manufacturer Bose used for their products to help identify irregularities in frequency response ([8], p. 80). Although, as with much of Lucier's work, there is a theme of interrogating acoustic space, it is clear that this interrogation is electromechanical in its nature and, as we have seen, the loudspeaker and its extensions were a central part of Lucier's vision for the piece.

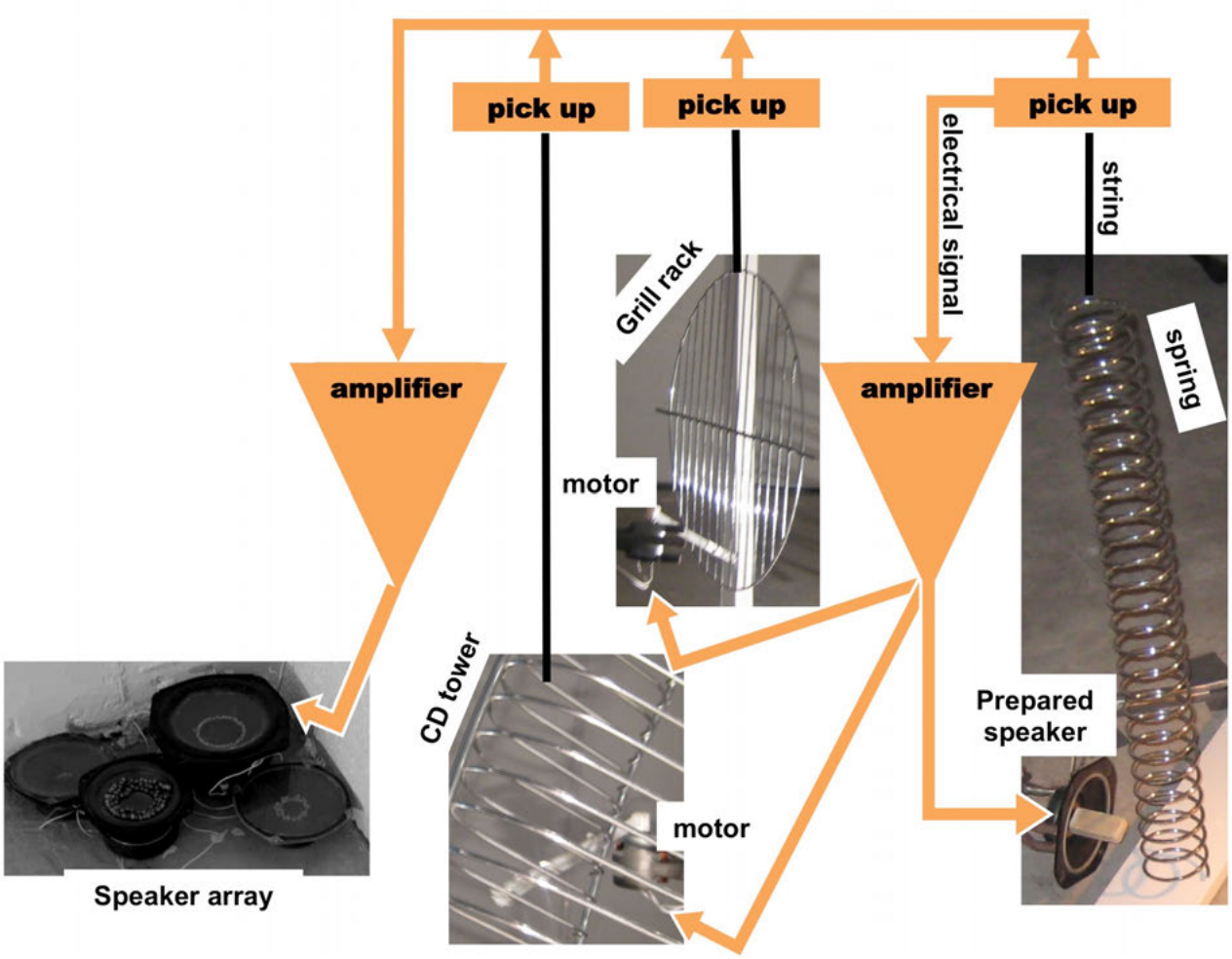


Figure 1. Technical scheme of Infinite Spring

It is this use of loudspeakers in the exploration of sympathetic vibration and resonance that makes this work relevant to my own creative experiments with loudspeakers and other electromechanical devices. Where Lucier has used percussion instruments for his speakers I have experimented with springs, foil, beads and plastic.

Other works of interest, from the same era, include Steve Reich's *Pendulum Music* (1968), which explores the musicality of electromechanical feedback via loudspeakers and swinging microphones. This piece is relevant as the resonant nature of the feedback is defined by the electromechanical system of the speaker and microphone combination, and further modified by the mechanical, swinging of the microphones. *Pendulum Music* is a music born entirely of a dialogue between the electrical and mechanical energies within the system, and their points of transduction. The main source of sound in *Infinite Spring* is also created by an electromechanical feedback system as can be seen in fig 1.

Contemporary sound art practitioners exploring the sonic potential of the electromechanical include Zimoun [15] with pieces such as '*Swarm of Prepared Vibration Motors'* (2008) and '111 Prepared DC Motors' (2010). Prepared electromechanical devices populate much of Zimoun's work, which also includes the use of loudspeakers. Zimoun's web site describes the work as a 'rigorous reductionism of the means used to produce sound' and as exhibiting an 'electric, dynamic sense of disquiet' [15]. The technique of using the DC motor as a loudspeaker or sounding device of sorts is also something I have employed in *Infinite Spring*.

Peter Bosch and Simone Simons used oscillating motors driven by 'musical phrases' ([1], p. 106) to excite a large sprung structure of shipping crates in *Krachtgever* (1998). Here, each crate is filled with a different material and the various resonances of the structure are explored by varying the frequency of the signal driving the motors. Much of Bosch and Simons work concerns itself with electromechanically induced vibration and resonant behavior, including '*The Electric Swaying Orchestra'* (1993) which, they claim, has much in common with *Pendulum Music* ([1], p. 104*)*.

Another contemporary practitioner worth considering here is Pierre Bastian [10] who has used motors as the driving force of his *Mechanium* mechanical orchestra and other installations since the 1970's. Here we find the electromechanical device in a context of traditional automata, and there are many themes relating this approach to my own, and to the other pieces considered above.

2.2 Music Technology Context

Music technology design has historically made use of electromechanical transfer for sound processing where purely electronic techniques were yet to be developed. Examples include the Leslie rotary speaker cabinet and the Ondes Martineau, which offered a choice of speakers, one of which vibrated a gong, whilst another had twelve strings strung over the front and back of the speaker to induce sympathetic vibration [8]. Early reverberation devices such as the spring and plate reverb used electromechanical transduction and amplification to extract sonorities from material objects, in a

process not dissimilar to that used by Cage in *Cartridge Music* (1960) [8], Stockhausen in *Microphonie* (1964) [8] as well as Tudor in *Rainforest*. Electromechanical user modifications have also appeared in this context, and there is much lively internet debate as to which rock guitarist was the first to purposely damage the speaker cone of their guitar amplifier to achieve a fuzz guitar tone and which recording was the first to feature electromechanical feedback from an electric guitar [4].

2.3 Towards a Theoretical Context

In these examples we find a balance of creative focus (in music, sound art and technological design terms) between the electronic domain of signal manipulation and the mechanical domain of material vibration. This is in contrast to what has become a more common creative approach in technologically mediated music, of focusing almost entirely on the realm of electronic signal manipulation, be it in the analogue or digital domain. It could be argued that the current prevailing model for music technology systems is one that views the loudspeaker as a subservient device, a mechanical 'limb' controlled by an electronic 'brain', whose resonant irregularities are problems to be eradicated wherever possible. Certainly in terms of computer based composition and synthesis, studio recording and production and electronically mediated live performance this is a broadly applicable model. Barry Truax ([13], p. 9) commented in 2001 that "it is significant that that the current emphasis of audio technology is almost entirely on the signal processing aspects and not the actual points of energy transfer". The prepared loudspeaker and other extended electromechanical interfaces seek to redress the balance of creative focus between electrical and mechanical energy in music and music technologies, and in doing so they reassert their power as the final gatekeeper in the signal chain of the electronic music system.

The prepared speaker achieves this through a method and practice that could be aligned to the discourse on *cracked media* as described by Caleb Kelly [9]. Primarily concerned with the breaking, scratching and purposeful creative destruction of sound recording media (vinyl, CDs etc) for sonic effect, the world of *cracked media* concerns itself with the physical user intervention of electric media. Kelly's book presents this partly as a critique of the recording, commercial music and technology industries as well as in the context of modern tactical and creative uses of everyday technologies. Whilst the prepared speaker's means may be radical and involve destructive user modification, we have seen how other technologies such as the Ondes Martineau and the mechanical reverb have achieved an electromechanical balance through the electronic design limitations of their era. An aesthetic and musical focus on electromechanical technology may therefore usefully be aligned to a notion of *media archaeology*. This field proposed by Singfried Zielinsky, among others, proposes that we 'find something new in the old' with regard to media technology, as opposed to accepting the 'continual march of progress' ([14], p. 3) towards new technologies. It sets out to 'destroy the Whig version of technological history' [12]. The loudspeaker is old technology; its basic moving coil design still in use today can be traced back to at least the 1890's ([2], p. 39). Many would argue it is a far from ideal technology with sonic irregularities that we are stuck with until better solutions are sought. The prepared speaker and other appropriations of electromechanical devices may offer new creative possibilities with old technologies in a practice that reflects the values of media archaeology.

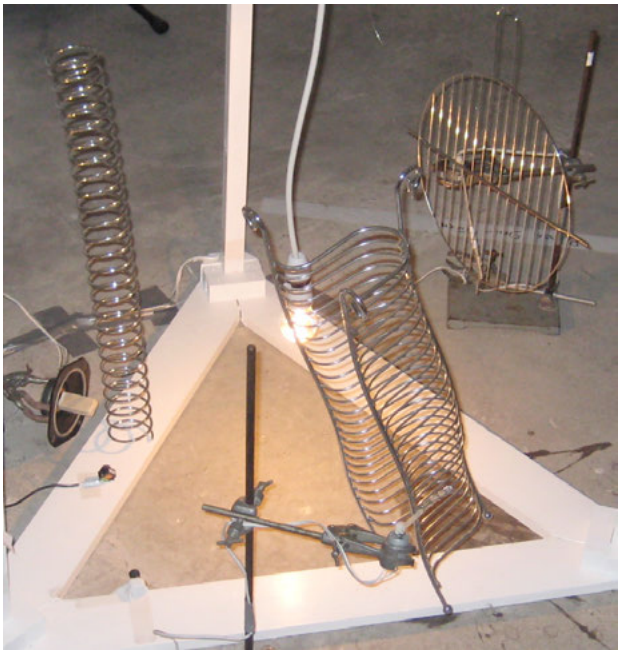


Figure 2. Infinite Spring installation at BV Studios 2010.

3. INFINITE SPRING

3.1 Technical Description

At the centre of my piece *Infinite Spring* is a prepared speaker system that has been developed to operate as an electromechanical oscillator (see fig 1). It is built from a large steel spring suspended such that its lowest coil touches a piece of wood that is glued to the centre of a loudspeaker (see fig 3). The top of the spring is coupled to a Piezo ceramic pick up which is used to transduce any movement or vibrations present into an electrical signal. The signal from the Piezo pick up is amplified and fed back to the loudspeaker whose cone is mechanically coupled to the bottom of the spring as described above. The vibration of the speaker sets the spring vibrating which, via a feedback chain of transduction and amplification, causes the speaker to vibrate (see fig 1). The whole electromechanical system oscillates when power is applied to the amplifier and it is possible to set the parameters so that it produces an audible, harmonically rich sound without causing uncontrollable feedback.

The signal from this spring / speaker oscillator is also used to drive two small DC motors. The DC motors here are essentially being regarded as rotary loudspeakers in a technique described both by Nick Collins [3] and the 'Electronic Peasant' [6]. With an AC audio signal driving them, the motors twitch and vibrate rather than rotate fully in their usual manner. The motors have armatures attached to their rotors that strike other metal objects in the installation (a metal CD storage tower and a grill tray – see figs 1 and 2), and the sound from these objects is also picked up using Piezo ceramic transducers, and amplified. The signals from the spring, the CD tower and the grill tray are amplified and routed out to a combination of three loudspeakers, one of which is prepared with beads, whilst another has the paper cone removed and replaced with a small tin cup (fig 4). The electromechanical sounds generated by the spring are modified by the electromechanical system of the motors, the CD tower and grill rack. All these sounds are further modified by the prepared speakers that broadcast the final mix in the room, and of course by the room itself. Figure 1 shows the full technical scheme for the piece and figure 2 shows an overview of how it was realised in its first public

exhibition at the BV open studios event in Bristol, UK, in 2010. Film documentation of the piece at this exhibition is viewable at www.sonicmarbles.co.uk.

Figure 3. Prepared speaker vibrating spring

3.2 Sonic Behaviour

Infinite Spring is a simple, dynamic system that produces a controlled but diverse and unpredictable music that is timbrally rich, always changing and never ending. The spring itself produces very deep tones, which would be inaudible without the contact microphone and amplifier system used to capture them. When this tone is used to drive the motors they twitch and vibrate sporadically creating a percussive element to the piece as the armatures strike the other metal objects. When the spring's tone is heard through the foil cup speaker it is diminished to a small tinny rattle.

Truax states that '…the electroacoustic process is not merely a simple extension of the capabilities of sound, but rather a fundamental transformation of how it works...it permits totally new concepts to operate' ([13], p. 124). At each point of transduction in *Infinite Spring* these 'new concepts' are able to operate. The amplification of the normally inaudible low frequencies of the vibrating spring, and the transference of those vibrations to other materials and objects, in such a way as *they* are made to sound in a form of mechanical synthesis, constitute Truax's transformations in this context. Even causing objects to sound continuously over long periods of time is a difficult thing to achieve in the purely mechanical world, yet easily achievable in the electromechanical one. The spring in this piece will vibrate continuously, or until someone switches the power off. Both Truax [13] and R. M. Schaffer [11] consider the dynamic behaviors of mechanical sound compared to the fixed waveform behaviors and immortality of electrical sounds. Infinite Spring creates a balanced dialogue between those two behaviors in a sonic and kinetic installation that draws attention to the physical cause and effect of sound as it is transmitted mechanically and electrically. This is achieved through the appropriation and modification of existing technologies, and through a design technique that uses nothing more complicated than would be found in an electrically amplified gramophone.

4. CONCLUSIONS

This paper has shown the context in which my piece *Infinite Spring* sits both in practical and theoretical terms, and in doing so has begun to set out an arena of the creative application of electromechanical transduction within the sonic arts.

Figure 4. Foil cup prepared speaker

It has exposed many themes and techniques that are worthy of continued exploration within creative sound and music practice, both in my own work and the work of others. This exploration will need to involve further surveys and analysis of works, practices and technologies, as well as practical experimentation. *Infinite Spring* is worthy of further practical development, particularly with regard to maximising the visual impact of the piece in any future exhibition scenario.

5. REFERENCES

[1] Bosch, P. and Simons, S. 2005. Our Music Machines. *Organised Sound* 10(2): 103-110.

[2] Chanan, M. 1995. *Repeated Takes*. London: Verso.

[3] Collins, N. 2006. Handmade Electronic Music: The Art of Hardware Hacking. Oxford: Routledge.

[4] DIY stomp boxes forum, accessed 2011: *http://www.diystompboxes.com/*. Torn speaker fuzz thread: http://www.diystompboxes.com/smfforum/index.php?action=printpage;topic=67452.0

[5] Driscoll, J. and M. Rogalsky. 2004. David Tudor's Rainforest: An Evolving Exploration of Resonance. *Leonardo Music Journal*, 14: 25-30.

[6] Electronic Peasant web site: *http://www.electronicpeasant.com/*

[7] Hamburg, T. interview with David Tudor, 1988. Available from http://www.emf.org

[8] Holmes, T. 2002. *Electronic and Experimental Music.* London: Routledge.

[9] Kelly, C. 2009. *Cracked Media: The Sound of Malfunction*. Cambridge, Massachusetts: MIT Press.

[10] Pierre Bastien web site: *http://www.pierrebastien.com/*

[11] Schafer, R. M. 1977. The Soundscape: our sonic environment and the tuning of the world: Vermont: Destiny

[12] Sterling, B. 2008. The Life and Death of Media. In: Miller, P.D. (ed). 2008. *Sound Unbound: Sampling Digital Music and Culture*. Massachusetts: MIT Press. Ch 6.

[13] Truax, B. 2001. *Acoustic Communication*. Westport CT: Greenwood.

[14] Zielinski, S..2008. Deep Time of the Media: Towards an Archaeology of Hearing and Seeing by Technical Means. Massatusits: MIT press.

[15] Zimoun web site: http://zimoun.ch/

Automatic Rhythmic Performance in Max/MSP: the kin.rhythmicator

George Sioros
University of Porto (Faculty of Engineering)
and INESC - Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
gsioros@gmail.com

Carlos Guedes
University of Porto (Faculty of Engineering)
and INESC – Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
cguedes@fe.up.pt

ABSTRACT
We introduce a novel algorithm for automatically generating rhythms in real time in a certain meter. The generated rhythms are "generic" in the sense that they are characteristic of each time signature without belonging to a specific musical style. The algorithm is based on a stochastic model in which various aspects and qualities of the generated rhythm can be controlled intuitively and in real time. Such qualities are the density of the generated events per bar, the amount of variation in generation, the amount of syncopation, the metrical strength, and of course the meter itself. The kin.rhythmicator software application was developed to implement this algorithm. During a performance with the kin.rhythmicator the user can control all aspects of the performance through descriptive and intuitive graphic controls.

Keywords
automatic music generation, generative, stochastic, metric indispensability, syncopation, Max/MSP, Max4Live

1. INTRODUCTION
In this paper, we propose an approach for real-time rhythm generation based on a stochastic model. This approach contrasts with recent ones involving evolutionary methods such as genetic algorithms [1][2], cultural algorithms [3] or connectionist approaches [4]. In our approach, the algorithm produces a rather static output with slight variations due to the stochastic nature of the algorithm that is characteristic of a certain meter and metrical subdivision level defined by the user. However, the output does not belong to a specific musical style. It is up to the user to modify and control the output of the algorithm during a performance by altering descriptive musical parameters that produce perceivable changes in the output such as the density of events per bar, the amount of syncopation, the degree of metrical strength, the amount of variation in generation, and of course the meter itself. In this sense, the algorithm behaves like a musical companion that responds musically to requests made by the user in musical terms.

kin.rhythmicator is built around two Max/MSP [5] externals (kin.weights and kin.sequencer) that implement the algorithm. It exists as a Max/MSP bpatcher and as a Max4Live [6] device.

2. THE ALGORITHM
The algorithm has two distinct phases. First, the meter entered by the user is subdivided into the number of pulses of a specified metrical subdivision level. Each pulse is assigned a weight value according to its importance in the meter so that a pattern characteristic of the meter emerges. In the second phase, the weight values are used to generate a stochastic performance.

These values are processed and mapped to probabilities of triggering events and their amplitudes in order to enforce or weaken the metrical feel, syncopate according to the specified meter and control the variations in the generated rhythm. The user controls these values indirectly through graphic controls. This gives a very intuitive control over these parameters and over the real-time rhythm generation. In the upcoming sections we describe in detail the steps taken to achieve these results.

2.1 Calculating the Weights
The calculation of the weights of the pulses is articulated in two phases: sorting the pulses by metric indispensability according to Clarence Barlow's metric indispensability formula [7] and calculating the weights based on the stratification levels.

These weights can be thought of as a measure of how much each pulse contributes to the character of the meter. A direct mapping of the weights to probabilities of triggering events gives rise to simple rhythmic patterns expected for the given meter. Variation in the performed rhythms is an innate quality of the algorithm arising from the use of probabilities in the performance.

2.1.1 Sorting by Metric Indispensability
The user inputs meter information in the form of a time signature and a metrical subdivision level which defines the number of pulses the measure is divided into – e.g. a 3/4 meter at the 16^{th} note metrical subdivision level has 12 pulses. Based on this information the meter is stratified by decomposing the number of pulses into prime factors (see Figure 1). Each prime factor describes how each stratification level is subdivided. The stratification level at index 0 is always a whole bar (prime factor 1). Different permutations of the prime factors describe different metrical hierarchies distinguishing this way between simple and compound meters like 3/4 and 6/8 – although they contain the same number of subdivisions at the sixteenth-note level (12) the first is decomposed as 1x3x2x2, while the second as 1x2x3x2.

Barlow´s indispensability [7] takes the prime factors of each stratification level and sorts the pulses in the meter according to how much each pulse contributes to the character of the meter, from the most indispensable to the least important.

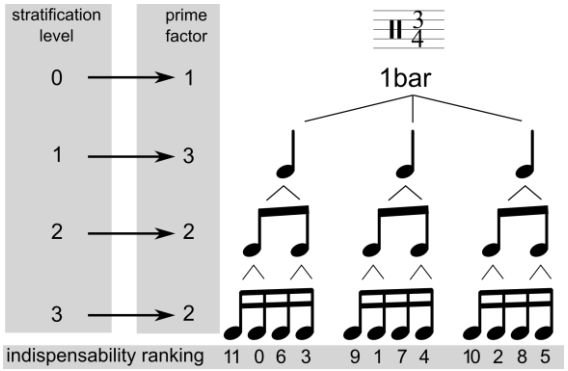


Figure 1. Stratification of a 3/4 meter to the 16th metrical level. At the bottom, the ranking according to Barlow's metric indispensability formula is shown.

2.1.2 Calculating the Weight Based on the Stratification Level

We assign to each pulse a weight based on the stratification level it belongs to and its indispensability ranking. Each level i has its own distinct range of weights W_i (see Figure 2):

$$W_i(\max,\min) = \left(R^{i-1}, R^i\right) \quad (1)$$

where R is a parameter related to the density of events of the resulted performance and ranges between 0 and 1. Equation (1) implies that the calculation of the ranges begins with the highest stratification level for i = 1 and continues until it reaches the metrical level defined by the user.

The pulse with the highest ranking value in each stratification level, i.e. the most indispensable, is assigned the maximum weight corresponding to the stratification level. The rest of the pulses in the stratification level are assigned smaller weights in the same range following a linear distribution. According to equation (1), for R = 1 all pulses have a weight equal to 1, while for R = 0 only the 1st stratification level survives.

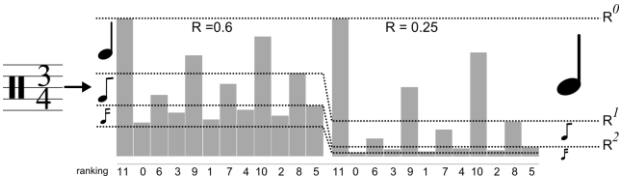


Figure 2. Weights calculated for a ¾ meter stratified to the 16th note level. The ranking of the pulses according to Barlow's formula is indicated below the assigned weights.

2.2 Stochastic Performance

Once the weights of all the pulses are calculated, a performance is generated by cycling through the pulses comprising the metrical cycle and deciding if an event will be triggered in each position or not. During performance, several aspects pertaining its style can be specified, such as the amount of syncopation, the density of events, the metrical strength, the amount of variation, and the events' articulation (staccato or legato).

2.2.1 Triggering Events

The probability of triggering an event on a certain time position is derived by the corresponding weight according to a simple exponential relation:

$$p_\ell = n \cdot {W_\ell}^M \quad (2)$$

where W_ℓ is the weight assigned previously to pulse ℓ, n is a normalization factor, and M is a user defined parameter related to the metrical feel and ranging between 0 and 1. The above equation functions as a "probability compressor", where for values of M close to 0, the differences in the probabilities are

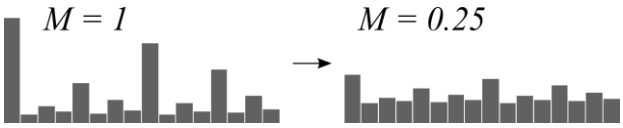


Figure 3. Probabilities are exponentially scaled.

smoothed out, while for values close to 1, the original probabilities arise (see Figure 3).

The amplitudes of the triggered events are calculated independently from the probabilities. They are directly proportional to the pulse weights at the strongest metrical feel.

2.2.2 Generating Syncopation

Syncopation is introduced in the generated rhythm by "anticipating" pulses in stronger metrical positions. Events are triggered according to the probability assigned to the immediately following next pulse. At the same time, the amplitudes are also anticipated, so that the amplitude of a syncopated pulse sounds louder, thus creating a dynamic accent. The user controls the probability P_S of anticipating a pulse which gives control over the amount of syncopation in the resulted rhythm.

Restrictions are imposed in order for the generated result to be more musical. A mechanism forces syncopation to stop when too many consecutive pulses are anticipated; otherwise for values of P_S close to 1 the resulted rhythm would be just an offset version of the non-syncopated one. An "off-beat" syncopation effect is achieved by resolving consecutive anticipated pulses to the next stressed pulse. Moreover, when only a couple of pulses are anticipated, an event triggered on the following stressed pulse would weaken the feeling of syncopation. In this case the stressed pulse is muted and will not trigger an event, independently from the corresponding probability.

2.2.3 Controlling Density

The density of events D refers to how many events are triggered per cycle. On average this is equal to the sum of the probabilities in all pulses:

$$D = \sum_{\ell=\text{all pulses}} p_\ell \quad (3)$$

The density of events and the metrical feel are by nature interrelated. This can be easily seen in extreme cases such as when the density is zero. Zero density means that no events are triggered which is, by definition, a non-metrical state. This degenerate rhythm could belong to any meter and tempo. Similarly, the metrical feel is weakened when events are triggered on every pulse, in other words when the density is maximum, and thus the meter can only be inferred from the amplitudes of the triggered events.

The density of events can be controlled by the parameter R in equation (1). Although the value of R cannot be used as a measure of the actual density of events it serves as an effective way of controlling it without affecting the metrical feel. The probabilities are distributed to the pulses taking into account the stratification level they belong to, preserving the hierarchy and structure of the meter even for low values of R, keeping this way a strong metrical feel when the density is low. On the other hand, the amplitudes of the triggered events are not affected by the changes in the parameter R. This way, when the density reaches its maximum (R = 1) the character of the meter is made evident by the amplitudes of the triggered events.

2.2.4 Controlling Metrical Strength

The strength of the metrical feel depends, on the one hand, on the probabilities assigned to the pulses and, on the other hand,

on the amplitudes of the generated events. A sense of meter is established when the events are triggered in important pulses (the most indispensable ones). The way the weights are calculated ensures that the more important a pulse is, the more often an event will be triggered in that position and this event will have an higher amplitude accordingly. The more the indispensability relation is preserved among the pulses, the stronger the metrical feel is. When all pulses have similar probabilities of triggering events and the amplitudes of the triggered events are random, not organized and do not establish a pattern, the resulted rhythm sounds random, not belonging to a specific meter.

In order to effectively control the strength of the metrical feel, the probabilities and amplitudes of the triggered events need to be adjusted simultaneously. The probabilities can be directly manipulated through the exponent *M* in equation (2). The normalization factor *n* ensures that the density of events *D* is not affected by the changes in the exponent *M*. In order to weaken the metrical feel as the value of *M* decreases, the amplitudes also get randomized but in a way that the distribution of amplitudes over time is kept constant.

Figure 4 summarizes the main aspects of the performance and their relation to the parameters of the algorithm.

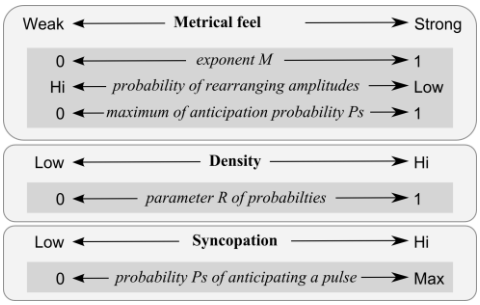


Figure 4. A summary of the basic user controls and the corresponding parameters in the algorithm.

2.2.5 Generating Variation

The generated rhythm varies and is non-repetitive due to its stochastic nature. The amount and type of variation can be controlled by restricting the mechanisms described above, namely the triggering of events and their syncopation.

At each pulse, two different decisions are made. First, it is decided whether the pulse will anticipate the next one according to the amount of syncopation set by the user. Second, the triggering of an event is decided according to the probability of the corresponding pulse or the following one when anticipating. The variation in the resulted rhythm is controlled by restricting the number of such decisions that are allowed to change from one cycle to the next.

Two modes of variation have been implemented: the stable and the unstable. In the stable mode, the variation revolves around an initial pattern which is randomly generated. In the unstable mode, the rhythm departs from an initial pattern and follows a random walk. It evolves constantly into new patterns. An initial pattern is always generated at the beginning of the performance but the user can re-generate a new random pattern at any time, creating an abrupt change in the performance.

2.2.6 Events' Articulation

The duration of the triggered events can be either fixed, in staccato mode, or can extend until the triggering of a new event, in legato mode. Syncopation is enhanced in legato mode by favoring the release of held events on stressed pulses even when no new event is triggered.

2.3 Controlling the Performance: the complexity space

The metrical feel, the amount of variation and the amount of syncopation form what we call a "space of complexity". A rhythm is considered to be simple, when the metrical feel is strong, variation is kept to a minimum and there is no syncopation. On the other hand, when the metrical feel is weak or when syncopation is introduced into the rhythm or when the rhythm is constantly changing, then the rhythm is perceived to be more complex. Rhythmic complexity in this sense is attributed to combinations of different aspects of the rhythm: metrical strength, syncopation and variation.

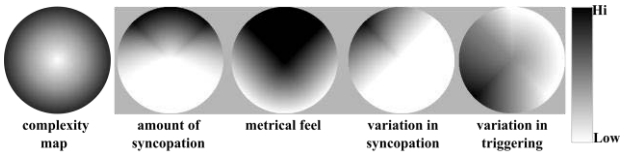


Figure 5. Contour plot of the functions used in the complexity plane to map position coordinates to the parameters of the algorithm. At the left side a contour of the expected complexity of the generated rhythms is shown.

We grouped the parameters of the algorithm related to complexity into a two-dimensional map (see Figure 5). As one moves away from the center the resulted rhythm becomes more complex. The dependence of each parameter on the position in the complexity map was empirically set, taking into consideration some basic restrictions derived from the nature of these parameters and our experience with various settings of the algorithm. Some of these restrictions are: i) when the metrical feel is low, syncopation is meaningless, ii) variation in the syncopation decisions apply only when the amount of syncopation is above a certain value, iii) when the amount of syncopation is significant the syncopation feeling is weakened by too much variation in the triggering decisions.

3. APPLICATIONS

3.1 Max/MSP Externals

The algorithm was implemented as two Max/MSP externals. Several other externals and abstractions have been developed in order to facilitate the use and implementation of the algorithm into Max/MSP applications. All externals and abstractions are completely cross platform, Windows and Mac OS.

The first phase of the algorithm, namely the generation of weights, is performed by the kin.weights external. The parameter *R* of equation (1), which controls the density of events, is directly fed into the external as a floating-point number in the range [0, 1].

The second phase, the triggering of events based on the parameters mentioned in the previous section is performed mainly by the kin.sequencer external. The weights calculated by kin.weights are fed into kin.sequencer which generates a performance by cycling through each pulse comprising the metrical cycle and deciding if an event will be triggered in that position or not. The amount of syncopation, the metrical strength, and the amount of variation can be controlled by respective messages to the external.

A java script suited for the jsui Max/MSP object was developed to visualize and improve user interaction with the complexity space described in 2.3.

3.2 The kin.rhythmicator bpatcher

The kin.rhythmicator Max/MSP bpatcher abstraction was built around the above externals. It is intended to be used in Max/MSP based applications and installations which

implement some kind of rhythmic interaction. Such installations can take the form of virtual musical instruments, compositional tools or interactive installations. It is easily integrated into Max/MSP patches. It can be controlled by various devices, from simple MIDI controllers to complex game controllers and is ready to directly trigger sound on any MIDI enabled synthesizer.

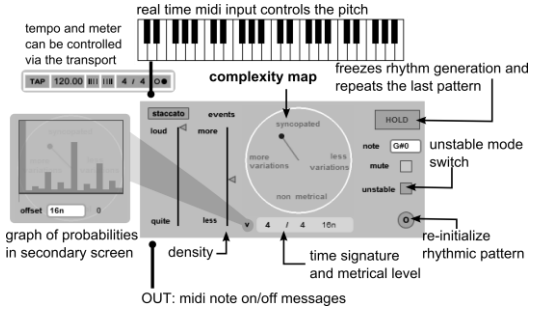


Figure 6. The interface of the kin.rhythmicator application.

The abstraction implements all the features of the algorithm and has a compact user interface when loaded into a bpatcher object (see Figure 6). All parameters of the algorithm can be set during performance directly on the user interface, as messages, or through the pattr system for storing preset files in Max/MSP.

Single notes or chords can be fed to kin.rhythmicator in real time making it follow a melody or a chord progression which can be either pre-scheduled, performed in real time by a musician or generated by some generative or analysis algorithm.

Several instances of the abstraction can be loaded at the same time for generating several rhythmic layers. All instances can be synchronized by the Max global transport. Also, one can generate polyrhythms by synchronizing different instances of kin.rhythmicator with different time signatures to the global transport.

3.3 The Max4Live MIDI Device

We developed a Max4Live device that can be used as a compositional and/or performance tool to dynamically generate rhythms. All parameters can be controlled through MIDI, automated with envelopes and saved together with the Live Set.

The device is built as a MIDI FX device, which means it can be loaded into a Live's MIDI track. It can be used alongside VST or the Live's instruments. More than one instance can be loaded in the same or different MIDI tracks. The interface is very similar to the Max/MSP bpatcher abstraction described above (see Figure 6).

The kin.rhythmicator max4Live device reads automatically the time signature and the play position of the Live transport and follows any time signature change in the song, so that there is no need to explicitly set the time signature on each kin.rhythmicator instance. All instances of the device are in sync with the rest of the Live Set. An offset parameter allows for a phase difference between each kin.rhythmicator and the global transport.

Two MIDI modes of operation have been implemented: thru and listening. In thru mode, the MIDI input is forwarded directly to the output without being changed. The generated rhythm is output as MIDI note on/off messages according to the MIDI note set on the kin.rhythmicator. In listening mode the rhythm generated follows the melody or chord progression at the input of the device.

4. CONCLUSION AND FUTURE WORK

The algorithm and applications introduced here present a novel approach to automatic rhythm generation. Departing from a preexisting metrical template containing the time signature and metrical weight distribution, and the subdivision level, a user can specify a performance controlling several musical parameters. Instead of specifying in detail the rhythmic parts and variations needed in a musical composition or performance, one can use kin.rhythmicator devices to control parts or the whole of the rhythmic section. These parts can be thought of as constrained improvisations that take the place of a detailed music score.

The real time and intuitive character of the controls and performance of the kin.rhythmicator helps in creating music more responsive to user actions. Controlling the metrical strength and density of events effectively has been made possible by taking into account the hierarchical structure of the meter in mapping the output of Barlow´s indispensability formula to the probabilities. A syncopation algorithm based on the anticipation of pulses that tends to keep a strong metrical feel is introduced.

Future development of the kin.rhythmicator algorithm and devices include the development of intelligent agents, which collaborate in generating a coherent output.

The kin.rhythmicator Max/MSP application and Max4Live device are available for download at our group website: http://smc.inescporto.pt/kinetic/

5. ACKNOWLEDGMENTS

This research was done as part of the project "Kinetic controller driven adaptive music composition systems", (ref. UTAustin/CD/0052/2008), supported by the Portuguese Foundation for Science and Technology for the UT Austin| Portugal partnership in Digital Media.

6. REFERENCES

[1] Bernardes, G., Guedes, C., Pennycook, B. "Style emulation of drum patterns by means of evolutionary methods and statistical analysis." *Proceedings of the Sound and Music Conference*, Barcelona, Spain, 2010.

[2] Eigenfeldt, A. "The Evolution of Evolutionary Software Intelligent Rhythm Generation in Kinetic Engine." *Proceedings of EvoMusArt 09, the European Conference on Evolutionary Computing*, Tübingen, Germany, 2009

[3] Martins, A. and Miranda, E. "Breeding rhythms with artificial life." *Proceedings of the Sound and Music Conference*, Berlin, Germany, 2008.

[4] Martins, A. and Miranda, E. "A connectionist architecture for the evolution of rhythms." *Proceedings of EvoWorkshops 2006 Lecture Notes in Computer Science*, Berlin: Springer-Verlag, Budapest, Hungary, 2006

[5] http://cycling74.com/

[6] http://www.ableton.com/maxforlive

[7] Barlow, C. "Two essays on theory". *Computer Music Journal*, 11, 44-60, 1987

Towards a Voltage-Controlled Computer Control and Interaction Beyond an Embedded System

André Gonçalves
Research Center for Science and Technology of the Arts (CITAR)
Portuguese Catholic University - School of the Arts
Rua Diogo Botelho 1327, 4169-005 Porto, Portugal
+351 916312148
hello@andregoncalves.info

January, 2011

ABSTRACT

The importance of embedded devices as new devices to the field of Voltage-Controlled Synthesizers is realized. Emphasis is directed towards understanding the importance of such devices in Voltage-Controlled Synthesizers. Introducing the Voltage-Controlled Computer as a new paradigm. Specifications for hardware interfacing and programming techniques are described based on real prototypes. Implementations and successful results are reported.

Keywords

Voltage-controlled synthesizer, embedded systems, voltage-controlled computer, computer driven control voltage generation

1. INTRODUCTION

In this paper I intend to share my realizations in developing an extended embedded system for Voltage-Controlled Synthesizers (VCS)[6] - the ADDAC system - which is an instance of what could be called a Voltage-Controlled Computer (VCC). One of the main objectives is to provide a platform that allows an easy integration of computer driven operations in a VCS.

This project started in 2008 as a concept and has been under research and development since then motivated by the nonexistence of such a system. It underlines and stresses the idea of the Voltage-Controlled Synthesizer as a unique instrument with its specific characteristics, as described by Chadabe, *the voltage-controlled synthesizer is not a simple object. It is a hardware system that is different in many ways from computers and from many other devices or systems that are also referred to as synthesizers*[1].

2. BACKGROUND

Analog synthesizers continue to be used by many musicians because of their distinctive timbres, intuitive real-time control and flexible patching[2].

Beyond what would be expected, voltage-controlled synthesizers are stronger today than they ever were, this is reflected on the amount of brands on today's market[1][8][7]. This tendency overcomes predictions that after the 1980's such analogue devices would fell under the appearance of the digital synthesizers. In fact conversely to what was expected, nowadays we witness an inverted trend, as these (analogue) devices are only getting stonger. By the end of the 80's a few new companies started to emerge and new users were captivated to the analog world of modular synthesizers. This tendency seems to cross all musical forms, not being specific of any particular genre.

New manufacturer's, following the pioneer's tradition (Hugh Le Caine, Harry Olsen, Raymond Scott, Bob Moog or Don Buchla), are also musicians or enthusiasts with musical background and/or interests. They are the ones who are responsible for bringing a renewed popularity as well as new paradigms into the VCS field. One denotes easily that each of these brands' have a genuine and devoted interest in creating such systems, an interest that goes way beyond business revenues expectations.

2.1 "Hello World"

Throughout the last 20 years, but specially in the last ten, the use of digital components in analog synthesizers became more common and there's now a fair amount of digitally driven voltage-controlled modules that integrate microcontrollers or digital signal processors in its circuitry. Still, none of these offer any I/O communication protocol to an external digital platform, they are closed in their software and controlled only by their analog inputs and panel controls, e.g. knobs and switches.

3. THE PARADIGM SHIFT

The historical evolution focusing on the methods used for the integration of external digital devices (MIDI devices and personal computers) with the VCS.

3.1 Control Voltage

Control Voltage (CV) is the standard name adopted to refer to the voltage source signals that are used to operate a VCS. CV generation is made by specific synthesizer modules, e.g. LFO's, envelopes (ADSR's), sequencers and noise sources.

3.2 MIDI

Since the 80's different MIDI to CV modules have been developed to allow the new digital world to interact with analog systems. These modules were designed in order to open the possibility of connecting the new digital synthesizers

[1] `http://wiki.muffwiggler.com/wiki/List_of_Modular_Synth_Equipment_Manufacturers`

NIME'11, 30 May–1 June 2011, Oslo, Norway.

with MIDI output to the VCS, converting MIDI messages (Note and Velocity) to a relative constant voltage source. This voltage source could then be used to control different modules fuctions, for ex. to drive the frequency of a voltage-controlled oscillator (VCO) at quantized notes. These interfaces are still today's standards. Most of them are installed in the synthesizer cabinet, side by side with other modules, and powered from the VCS power supply.

The MIDI to CV and, later on, the CV to MIDI interfaces, act like a bridge (Figure 1) between any MIDI device and the VCS, this implementation also allows any computer to be connected through a standard MIDI interface and MIDI capable software. There's no processing done in these interfaces, their circuitry allows a linear conversion from a midi note to its relative constant voltage source at a standardized 1 volt per octave range. This analog to digital conversion uses 8 bit messages, where only 7 bits are used for note resolution, allowing a maximum range of 128 values. These interfaces are mostly monophonic, meaning that they only allows one note to be played at a time.

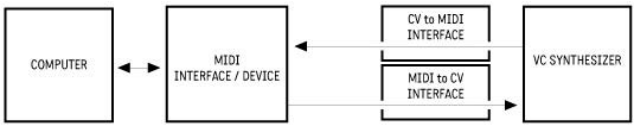


Figure 1: MIDI Diagram, an integrated system

3.3 VOLTA

In early 2009 MOTU released VOLTA, a virtual audio instrument, compatible with most audio software, that uses the outputs of an external sound card[2] to generate voltages. The voltages generated are defined by the track automation settings defined in the software. This only allow one way communication: from the computer into the VCS.

Likewise in MIDI communication, the sound card hardware acts like a bridge between the computer and the VCS, there's no processing done in the interface. The paradigm shift resides in the fact that the system does not intend to replicate the standard MIDI use, and not limit itself to solely translate notes to their relative frequencies, it goes beyond it. Using the sound card's, 16 bit resolution digital to analog converters, it allows a new range of possibilities, sweeps can now be effectively made and consequently the creation of LFO's, ADSR's, etc.

The aforementioned system requires that a computer node is present at all times, furthermore the necessary sound card is not integrated in the VCS cabinet which, besides affecting the ease of use, affects portability. Also regarding cabling, special cables are needed in order to connect both sides: mono jacks 1/4 inch to 1/8 inch.
(Figure 2)

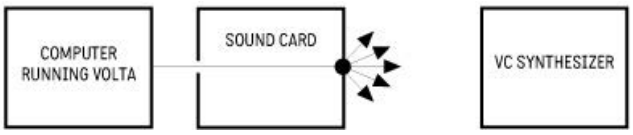


Figure 2: VOLTA to CV Diagram, a non-integrated system

[2]Not compatible with all soundcards, soundcard's outputs need to be DC coupled.

3.4 Conclusions

In my point of view, there's an underlining principle that can, in a first instance, effectively separate both approaches: integrated and non-integrated systems. Integrated systems like most MIDI to CV modules are installed in the VCS cabinet nad powered from its power supply. Non-integrated systems are peripheral devices not installed in the VCS cabinet and powered from their own power supply. This separation also highlights another distinction: the first approach is idealized from the synthesizer point of view, and the second from the computer point of view.

4. TOWARDS A VOLTAGE-CONTROLLED COMPUTER

By the end of 2009 I prototyped my first module (AD-DAC001 Brain Unit). The system was designed with very important features in mind:

1. For convenient usage, it was important that it was an **integrated system**, mounted in the synthesizer cabinet, side to side with all the other modules.
2. It would **not** be **computer dependent**.
3. If a computer is used, then the **communication would happen in two ways**, from the synthesizer to the computer and vice-versa.
4. Analog to digital and digital to analog **conversions** would have at least **16 bit resolution**.
5. It could be used in two ways:

   As a **"master" / standalone device**

   Or as **"slave device"** connected through one single usb cable straight to a computer

4.1 Definition

The Voltage-Controlled Computer (VCC) is an hardware based Embedded System locating itself in a specific spectrum of Embedded Computing (Figure 3) that also complies to the standards and tradition of the Voltage-Control Synthesizer as in [6][1][3].

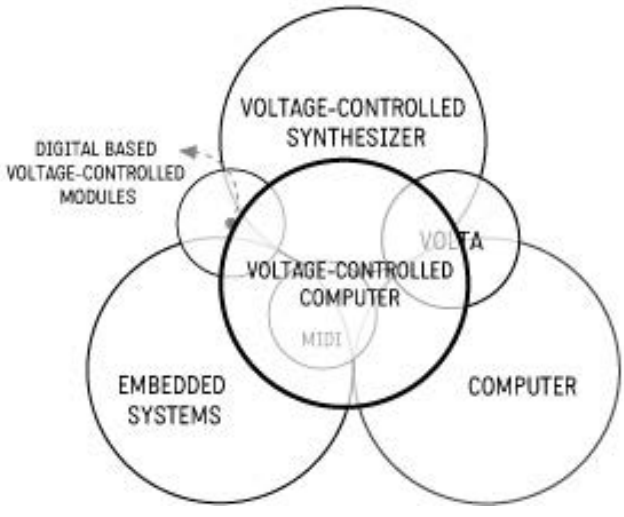


Figure 3: Positioning diagram

The VCC is not computer dependent.

The VCC creates a new paradigm in computer to VCS communication and interface. It no longer acts as a bridge between the digital world (computer or midi device) and the VCS, its microcontroller provide the autonomy and computational power to run complex algorithms that establish new interactions between analog and digital sources. (Figure 4)

Being a digital device, the VCC can be programmed to communicate to most digital platforms. These are regarded as peripherals that have specific functions that augment the possibilities and complexity of the system, e.g. a computer, MIDI device, mobile device, router, gamepad.

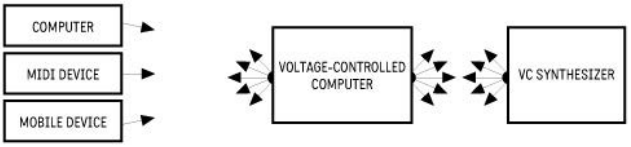


Figure 4: VCC Diagram, an integrated system

4.2 Standards

- The VCC complies with both Voltage-Controlled Synthesizer and Embedded Systems Design standards.
- The VCC analog inputs and outputs are converted to and from digital signal levels at standardized CV voltage sources. These Control Voltages operate at standardized VCS bipolar ranges.
- The VCC is comprised of one main module that hosts the microcontroller and optional expansion modules that expands both the system's possibilities and potential.
- The VCC connects to a standard VCS Bus Board and powers itself from standard bipolar power supplies.
- The VCC follows VCS standardized front panel dimensions.
- The VCC can communicate with the digital world through different standard communication protocols.
- The VCC trusts on its reentrant[5] software safety and reliability.

Figure 5: ADDAC001Brain Unit Prototype I & II, 2010

5. ADDAC SYSTEM

The VCC concept lead to the research and development of the ADDAC System[3]. This system has been developed in the last year and, due to public request, became a commercially available product.

5.1 Ground work

Most of the initial work was divided into two sections: the analog circuitry required to operate on bipolar voltages and the creation of dedicated software to run in the microcontroller.

The first Brain Unit prototype (Figure 5) resolved most of VCC's previously defined electronics specifications becoming a developer's platform for software programming and debugging.

5.2 The 00X System

The 00X System (Figure 6) is comprised of a Brain Unit and seven different expansion modules with specific functions that connect to the Brain Unit augmenting the interaction between the analog and digital worlds.

ADDAC00X System modules:

- ADDAC001 Brain Unit
- ADDAC002 CV / Manual inputs
- ADDAC003 Manual inputs
- ADDAC004 Gate inputs
- ADDAC005 Gate outputs
- ADDAC005W "Well tuned"[4]. gate outputs
- ADDAC006 Nunchuck input
- ADDAC007 Ethernet Input
- ADDAC008 Midi Input

Figure 6: The ADDAC System, March 2010

5.3 Technical Specifications

The overall system architecture is developed around the Arduino[4] open-source hardware and software platform. The Arduino C++ code framework is stripped to its core developer's environment, the heart of the Brain Unit operating system is integrated in the Arduino software as an external library.

The adopted microcontroller is an ATMEL ATMEGA1280 running at 16 Mhz. The analog circuitry is designed to allow inputs and outputs at bipolar -10/+10 volts range (standard CV voltage range). Instead of using the microcontroller's

[3] `http://www.addacsystem.org`

[4] Arduino is an open-source physical computing platform based on a simple i/o board, and a development environment for writing Arduino software
`http://www.arduino.cc`

dedicated analog and PWM pins, external, 16 bits resolution, Analog Devices converters (ADC's and DAC's) where also integrated in the schematic to maximize precision.

For USB communication an FTDI Serial to USB converter is used.

The system conforms to Eurorack[5] size format and use standard 3.5mm mono jacks for its physical inputs and outputs connections.

It also conforms to standard 8x2 pin Bus Board power connectors and -12 / +12 volts bipolar power supplies.

5.4 Front panel

ADDAC001 VS2. front panel features:

- 8 analog CV outputs
- 1 offset knob per output
- 1 led per output to monitor voltage state
- 2 Hex switch encoders to select pre-programmed presets
- 1 on-board knob to be assigned to any specific code function
- Reset Switch
- Midi input
- Nunchuck[6] remote input
- USB connector
- 2 led's to monitor Serial communication activity

5.5 Open-source C++ Framework

The C++ open-source framework defines most of the necessary setup for the software to operate properly, resolving most low level operations such as:

- Defining all specific pin I/O assignments, these are physical connections of the microcontroller's IC pins and were defined during the schematic development in order to have a clean pcb design.
- Facilitating classes to resolve most standard communication protocols, serial, MIDI, open sound control (OSC)[9].
- The integration of several algorithms like complex LFO's, linear and logarithmic ADSR's, Lissajous curves and complex randoms functions to mention just a few.

5.6 Software

There's four different possible ways for the system to communicate with a computer, these have different levels of technical know how required:

1. An open-source C++ library for advanced developers allows full access to the system's core software.
2. A set of open-source Max-Msp patches allows direct implementation, of pre-programmed classes, in Max-Msp
3. A standalone application with a dedicated GUI that integrates diferent protocols to interact, OSC, MIDI, Serial.
4. An Ableton Live audio instrument

6. CONCLUSIONS AND FUTURE DEVELOPMENTS

The system described brings new powerful tools to the use of the Voltage-Controlled Synthesizer, and follows its standards in voltage and operation method. It facilitates an integration with today's state of the art musical softwares, devices, controllers and programming frameworks be it through Serial, USB, OSC or MIDI. It provides a different and totally new method from the traditional MIDI to CV interfaces allowing new ideas to be developed due to its greater 16 bit precision range in converting digital to analog signals and vice-versa. It allows the user to rethink the computer interaction within an analog system facilitating new approaches, functions and integrations in an easy and user friendly way, not possible before with any commercially available module.

The system definitely fills a necessity I had in my analog modular system and has been the main control voltage source, putting aside most control voltage generators i had prior to this. It has also been used in most of my live performances proving to be one of the most important modules in my VCS. I've also been in close contact with the users who are already using it either discussing ideas and improvements, developing new functions or rethinking future hardware versions.

Future developments will mainly focus on two aspects that I find very important: Upgrading the CPU to a faster one, I realized that speed is the next issue that needs to be resolved, most probably to an ARM Cortex^{TM} Processor[7], leave the Arduino environment aside and re-program the whole platform in C; Develop a dedicated software application for computers and mobile devices that, through a Graphic User's Interface, will allow users to program the microcontroller without needing to know any code language.

7. ACKNOWLEDGEMENTS

Arduino Foundation, for the ground breaking work in open-source physical computing.
`http://www.arduino.cc`

Robin Price, for the ground breaking work interfacing an Arduino with an AD5668 16 bit DAC.
`http://registeringdomainnamesismorefunthandoingrealwork.com`

Jean-Philippe Lambert, for the initial help on parsing Serial communication in Max-Msp.

8. REFERENCES

[1] J. Chadabe. *The Development And Practice of Electronic music*, chapter The Voltage-Controlled Synthesizer. Prentice-Hall, Inc., Eaglewood Cliffs, New Jersey, 1975.

[2] A. Chaudhary. Band-limited simulation of analog synthesizer modules by additive synthesis. *Center for New Music and Audio Technologies University of California, Berkeley*, 2001.

[3] N. H. Crowhurst. *Electronic Musical Instruments*. TAB Books, 1971.

[4] K. Gann. La monte young's well-tuned piano, 1997.

[5] J. G. Ganssle. *The Art of Programming Embedded Systems*. Academic Press, Inc., Orlando, FL, USA, 1st edition, 1991.

[6] R. A. Moog. Voltage-controlled electronic music modules. *Journal of the Audio Eengineering Society*, 13(3), July 1965.

[7] G. Robair. *Something old, something new*, pages 46–62. Electronic Musician, 2001.

[8] G. Robair. *Analogue Renaissance*, pages 46–62. Electronic Musician, 2006.

[9] A. Schmeder, A. Freed, and D. Wessel. Best practices for open sound control. In *Linux Audio Conference*, Utrecht, NL, 01/05/2010 2010.

[5] Eurorack is one of today's most used VCS standard size format and follows the standard 19" Rack unit system measured in U's.

[6] Standard Nintendo Nunchuck Remote game controller.

[7] ARM Cortex^{TM} is a popular processor used in devices like the iPad

Polyhymnia: An automatic piano performance system with statistical modeling of polyphonic expression and musical symbol interpretation

Tae Hun Kim, Satoru Fukayama, Takuya Nishimoto and Shigeki Sagayama
Graduate School of Information Science and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
{kim, fukayama, nishi, sagayama}@hil.t.u-tokyo.ac.jp

ABSTRACT

We developed an automatic piano performance system called Polyhymnia that is able to generate expressive polyphonic piano performances with music scores so that it can be used as a computer-based tool for an expressive performance. The system automatically renders expressive piano music by means of automatic musical symbol interpretation and statistical models of structure-expression relations regarding polyphonic features of piano performance. Experimental results indicate that the generated performances of various piano pieces with diverse trained models had polyphonic expression and sounded expressively. In addition, the models trained with different performance styles reflected the styles observed in the training performances, and they were well distinguishable by human listeners. Polyhymnia won the first prize in the autonomous section of the Performance Rendering Contest for Computer Systems (Rencon) 2010.

Keywords

performance rendering, polyphonic expression, statistical modeling, conditional random fields

1. INTRODUCTION

We developed an automatic piano performance system called Polyhymnia. To our knowledge, it is the first system that is able to learn and predict polyphonic expression in piano music with diverse performance styles. Human prefer an expressive performance rather than a flat performance obtained by direct converting into MIDI format, and therefore computer-based tools for an expressive music performance would be useful for computer-aided music creations and performances. Unfortunately, automatic rendering of an expressive performance with a music score is a very difficult problem since expressive performance is one of the most complicated human tasks, and its mechanism is still not clear.

There exist many instruments for performing music. Since each instrument has different mechanical design, developing an universal method for automatic renditions of any musical instruments is extremely difficult. We are focusing on piano renditions since piano has abundant solo pieces so that it promises a useful application for computer-aided music creations and performances. Fortunately, musical expression in piano music can be represented with only 3 expression parameters: instantaneous tempo, loudness (velocity) and performed duration. Such a simple parametric representation allows us to develop a simple model of piano performance that can be well encoded in MIDI format.

Polyhymnia fully automates an expressive piano performance. Musical symbols provide a basic guideline for an expressive performance and they can be interpreted in several ways. Therefore we propose flexible parametric models for their automatic interpretation. Polyphonic features of expressive piano performance is very important since piano music is usually polyphonic. We discuss them in this paper and call musical expression with such features *polyphonic expression*. We proposed a statistical modeling of polyphonic piano renditions and showed that generated performances with polyphonic expression sounded better than performances without it [4]. We briefly describe the idea behind the proposed modeling and show how to implement it with Conditional Random Fields (CRFs).

An automatic piano performance system should be able to deal with various unknown piano pieces. Experimental results on performances generated by Polyhymnia with various compositions indicate that they had polyphonic expression and sounded expressively. A piano piece can be performed with diverse performance styles. One of the benefits of the proposed modeling is that diverse models can be easily obtained by training with various performance styles. Experimental results on diverse performances generated by Polyhymnia indicate that each trained model reflected the style observed in the training performance set.

Polyhymnia participated in the Performance Rendering Contest for Computer Systems (Rencon) 2010, and won the first prize in the autonomous section of the contest.

2. RELATED WORK

Several systems for automatic piano renditions are proposed [5]. Director Musices and the Rubato system utilize sets of performance rules extracted by music experts. Kagurame series and COPER are based on several searching algorithms from human performances. ESP, YQX and Usapi try to statistically model musical expression in piano music, whose parameters are learned from training performances. Most of those systems discuss renditions of monophonic melodies, and polyphonic renditions have not been well discussed due to computational complexity and necessity of a huge amount of data. In addition, automatic interpretation of musical symbols were not well discussed since they input a score in MIDI-like format, and it is based on very simple rules.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

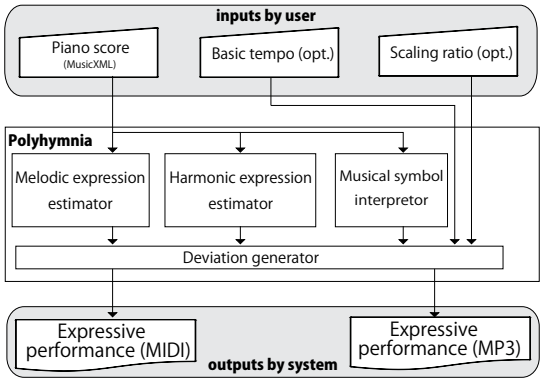


Figure 1: Polyhymnia architecture.

Some commercial notation softwares are also able to provide an expressive performance of a given piece. Although it is unclear how they generate musical expression, their methods are probably based on interpretation of musical symbols with simple rules.

3. SYSTEM OVERVIEW

To obtain an expressive piano performance with Polyhymnia, users requires only to input an piano score in MusicXML[1] format without any other configurations. Unlike MIDI format, MusicXML is able to encode almost all kinds of musical symbols digitally. Encoded musical symbols are automatically interpreted with parametric models that are flexible to generate various interpretations of each occurrence of a symbol. Polyphonic expression is learned and generated with Conditional Random Fields for polyphonic piano renditions. The depth of generated musical expression can be controlled by scaling ratios. The system provides expressive performances in MP3 and MIDI formats (Figure 1).

4. MUSICAL SYMBOL INTERPRETATION

4.1 Expression marks

Dynamic marks such as *p*, *mf* and so on, should be mapped to concrete MIDI velocity values. To find such mapping, 15 performances of V. Ashkenazy in CrestMuse PEDB [2] were analyzed. Table 1 shows the analytical result indicating that interpretation of each occurrence of a mark is distributed, and its interpretations in upper and lower staffs are different over all dynamic marks, for example, marks in lower staff are performed softer than those in upper staff. In order to interpret dynamic marks automatically, given marks should be mapped to concrete values with various maps. As a simple solution, Polyhymnia simply maps given marks to the estimated mean values for upper and lower staffs, respectively. However, this can be improved by proper selection of a map for each occurrence of a mark.

crescendo, *diminuendo* and *ritardando* should be interpreted with gradual changes of loudness and tempo. It is well known that human perceives them by exponential changes of sound energy and tempo in BPM. With analysis of human performances, we found that human performers performs such changes in various forms, and interpret *ritardando* with gradual decreasing tempo and loudness. To model such interpretations, we propose an parametric mathematical model for loudness and tempo changes. Let d_t be loudness in MIDI-velocity[2] or instantaneous tempo in log-

[1]http://www.recordare.com/musicxml

[2]MIDI-velocity can be regarded as a logarithmic scale for sound energy.

Table 1: Human interpretation of dynamic marks. Note that all averages and standard deviations are in MIDI velocity. *ppp* and *mp* were not occurred in the data.

Upper staff

	ppp	*pp*	*p*	*mp*	*mf*	*f*	*ff*	*fff*
Occur.	-	157	2087	-	67	1490	418	19
Avg.	-	50	52	-	58	67	76	98
St. dev.	-	14	15	-	8	15	16	2

Lower staff

	ppp	*pp*	*p*	*mp*	*mf*	*f*	*ff*	*fff*
Occur.	-	150	3169	-	53	1538	353	12
Avg.	-	37	37	-	47	57	73	101
St. dev.	-	13	11	-	9	20	19	11

BPM at time t. Then, its gradual changes over t can be modeled as

$$d_t = d_0(\beta \cdot t^{\alpha} + 1), \tag{1}$$

where d_0 is start value, β is the parameter for expression depths and α is the parameter for shapes. If α is 1.0, energy and tempo in BPM are changing exponentially. With different setting of α and β, each occurrence of a mark can be interpreted in various forms. As a simple solution, Polyhymnia interpret all occurrences of a mark with fixed parameter values. However, this can be improved by automatic determination of parameter values for each occurrence of a mark.

4.2 Ornaments

Mordent, *turn*, *trill* and grace notes are performed with additional notes. Since such additional notes decorate their parent notes, we can assume that their loudness is determined based on their parent note's loudness. However, human is not able to perform a note sequence with a constant velocity. Assuming that such motor error is following Gaussian distribution, loudness of the ith additional note d_i can be modeled as

$$d_i = d_0 + \mathcal{N}(0, \sigma^2), \tag{2}$$

where d_0 is the loudness of the parent note. σ^2 controls fluctuation ranges of loudness.

Arpeggio indicates that onset time of each arpeggiated note should be delayed one after another. Since human is not able to perform such notes with a constant delay, we can assume that it contains Gaussian noise. Then, onset time of ith arpeggiated note d_i can be modeled as

$$d_i = d_0 + i \cdot \Delta + \mathcal{N}(0, \sigma^2), \tag{3}$$

where Δ is a delay time, and d_0 is the onset time of the lowest arpeggiated note.

5. STATISTICAL MODELING OF POLYPHONIC PIANO RENDITIONS

5.1 Polyphonic expression in piano music

Although musical symbols provides a basic guideline for an expressive performance, musical expression in piano music is much more complicated, for example, instantaneous tempo, loudness and performed duration are fluctuating over time, even if there are no musical symbols for them. In addition, an expressive piano performance has polyphonic expression whose features include:

- Each voice expression has fluctuations of loudness and performed duration over time, and it is not always same to the other voices.

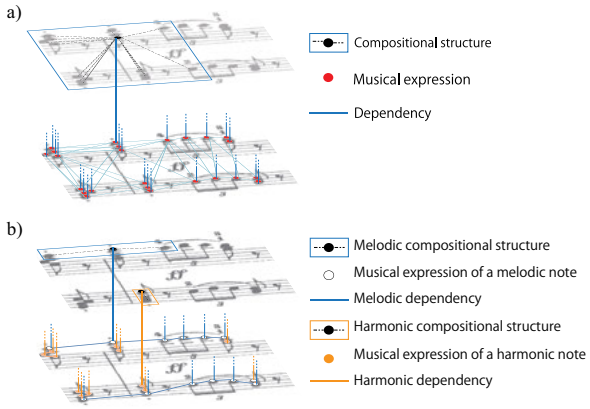


Figure 2: Complex dependency networks of polyphonic expression in piano music (a). Simplified dependency networks by introducing melodic and harmonic dependencies (b).

- Expression of each note in a chord is not always same to the other notes in the chords. Playing a chord with different combinations of note expression results different sounds of the chord.

In order to learn and predict polyphonic expression in piano music, we proposed a statistical modeling of polyphonic piano renditions and showed its efficiency for improving a machine-rendered piano performance [4]. In following subsections, we briefly describe the idea behind the modeling and show how to implement it with Conditional Random Fields.

5.2 Probabilistic formulation

Prediction of an expressive performance D given a piano score $\mathbf{S}$ can be formulated probabilistically such as

$$\hat{D} = \arg\max_{D} P(D|\mathbf{S};\Theta), \tag{4}$$

where Θ is the parameters of the distribution. To model $P(D|\mathbf{S};\Theta)$, we assume that a note expression is dependent on its compositional structure represented with score features and on the other note expressions. Figure 2a shows a dependency network of polyphonic piano music. In case of polyphonic expression, such dependency is very complex, and therefore it is hard to model it with computational tractability, and a huge amount of training data is necessary for learning model parameters Θ. Therefore, an approximation to polyphonic expression is necessary for a tractable modeling.

To simplify dependency in polyphonic renditions, we proposed an approximation with melodic and harmonic dependencies. Figure 2b shows an example of simplified dependency network with the proposed approximation. We believe that such approximation promises a perceptually best performance. This is because human perceives

- different voice expressions sounding simultaneously,
- different sounds of a given harmony,
- expressions of outer voices easier than that of inner voices [3].

Hence, $P(D|\mathbf{S};\Theta)$ can be approximated such as

$$P(D|\mathbf{S}) = P(D^{\mathrm{m}^u}|\mathbf{S}^{\mathrm{m}^u}) \cdot P(D^{\mathrm{m}^l}|\mathbf{S}^{\mathrm{m}^l}) \cdot \prod_{h^u=1}^{H^u} P(D^{h^u}|\mathbf{S}^{h^u}) \cdot \prod_{h^l=1}^{H^l} P(D^{h^l}|\mathbf{S}^{h^l}), \tag{5}$$

where $P(D^{\mathrm{m}^u}|\mathbf{S}^{\mathrm{m}^u})$ and $P(D^{\mathrm{m}^l}|\mathbf{S}^{\mathrm{m}^l})$ are distributions of melodic expression in the uppermost and lowermost voices, respectively, and $P(D^{h^u}|\mathbf{S}^{h^u})$ and $P(D^{h^l}|\mathbf{S}^{h^l})$ are distributions of harmonic expressions in upper and lower staffs, respectively.

Since such approximation allows Markov assumption on both of melodic and harmonic dependencies, they can be modeled with statistical models with hidden state transitions, such as Dynamic Bayesian Networks, Hidden Markov Models and Conditional Random Fields. Considering that our goal is to estimate a note expression sequence given a sequence of score feature vectors representing a piano score, we believe that CRF is one of the best frameworks for modeling those dependencies.

5.3 Modeling with Conditional Random Fields

We assume that a melodic compositional structure is represented with score features, such as pitch, duration, note interval and so on, and a harmonic compositional structures is represented with score features, such as pitch, duration and so on. Also, we assume that melodic expression is represented with instantaneous tempo, loudness and performed duration, and harmonic expression is represented with onset time differences, loudness and performed duration[3].

Although score features and expression parameters of melodic and harmonic dependencies are different to each other, they can be modeled with CRFs with the same model structure. Let d_n and D be the nth melodic or harmonic expression and its sequence, respectively. Let $\mathbf{S}$ be a sequence of score feature vectors representing melodic or harmonic compositional structures, and s_k be the kth score feature. Assuming that d_n is only dependent on d_{n-1} (Markov assumption), we can define the jth feature function F_j such as

$$F_j(D,\mathbf{S}) = \sum_{n=1}^{N} \delta(\{d_{n-1}, d_n, s_k\}_j, n), \tag{6}$$

where $\delta(\cdot)$ returns 1, if the jth triple from all possible triples of $\{d_{n-1}, d_n, s_k\}$ is occurred at position n, and 0, otherwise.

Introducing a weight variable θ_j for each F_j and according to the Maximum Entropy Principle, $P(D|\mathbf{S};\Theta)$ can be defined such as

$$P(D|\mathbf{S};\Theta) = \frac{1}{Z(\mathbf{S},\Theta)} \exp \sum_j \theta_j F_j(D,\mathbf{S}), \tag{7}$$

where

$$Z(\mathbf{S},\Theta) = \sum_{D'} \exp \sum_j \theta_j F(D',\mathbf{S}). \tag{8}$$

Model parameters Θ can be learned from training performances with Maximum Likelihood Estimation by an iterative algorithm, such as Stochastic Gradient Descent [1]. Once Θ is estimated, we can predict an expressive performance with equation (4), and this can be efficiently computed with Forward-backward algorithm and Dynamic Programming technique [6].

6. EXPERIMENTAL EVALUATION

6.1 Generation quality

An automatic music performance system should be able to render *unknown* pieces in various compositional styles. In order to evaluate Polyhymnia in this aspect, piano pieces in various compositional styles were rendered by the system, and evaluated by 19 human listeners[4]. We rendered

[3]Details of the melodic and harmonic score features and the expression parameters can be found in [4]

[4]2 professional musicians, 13 hobby musicians and 2 non-musicians participated in the listening experiments.

Table 2: Test pieces used in the experiment.

ID	Composer	Piece	Tempo
CF	Chopin	Mazurka no. 5, op. 7-1	fast
CS	Chopin	Sonata no. 2-3, op. 35	slow
MF	Mozart	Sonatina no.5-3, KV. 439	fast
MS	Mozart	Marche Funebre, KV. 453a	slow
RT	S. Joplin	The Entertainer (ragtime)	middle
GR	Grieg	7 Lyric Piec., 7. Rem., op. 71	slow

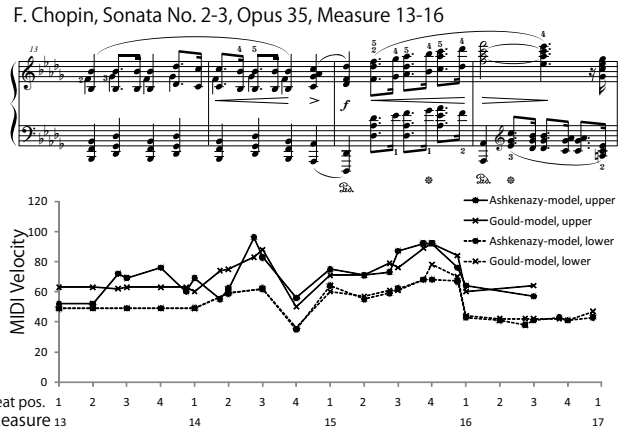


Figure 3: Generated performances by Polyhymnia: F. Chopin, Sonata no. 2-3, op. 35.

6 unknown pieces with Polyhymnia as shown in Table 2. Note that the test pieces included not only F. Chopin and W. A. Mozart's pieces, but also E. Grieg and S. Joplin's pieces whose compositional styles are quite different to the training pieces.

In order to render expressive performances with different performance styles, we prepared two different models such as *Ashkenazy-model* and *Gould-model* trained with 15 performances of V. Ashkenazy[5] and 7 performances of G. Gould[6], respectively (CrestMuse PEDB). We prepared 5 performances for each piece such as performance without expression (D), by musical symbol interpretation only (E), generated with *Ashkenazy-model* (S1), generated with *Gould-model* (S2) and by a human performer (H). All of those sound samples were blind to the listeners, and their human-likeness and musicality were evaluated using 6-level-scales.

Figure 3 shows an example of generated performances by Polyhymnia. The results indicate that the had polyphonic expression, and their fluctuations were different to each other. Figure 4a shows the average scores of the 6 test pieces. Analysis of Variance on those average differences with $p < 0.05$ indicate that performances generated by Polyhymnia sounded better than performances without expression. Score difference between S1 and H was not significant. This means that performances generated with *Ashkenazy-model* sounded expressively like human performances do. Score differences between S2 and H were not significant in some particular pieces. This means that some performances generated with *Gould-model* sounded expressively like human performances do.

[5]Prelude no. 1, 4, 7, 15, 20, Etude op. 10-3, 10-4, 25-11, Waltz op. 18, 34-2, 64-2, 69-1, 69-2, Nocturne no. 2, 10.
[6]Piano Sonata KV279-1, 279-2, 279-3, 331-1, 545-1, 545-2, 545-3.

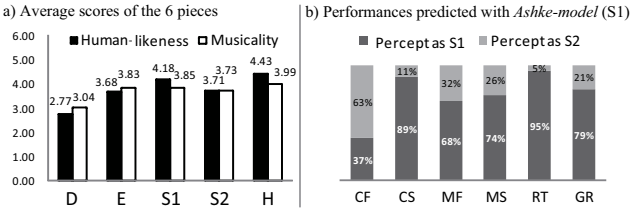


Figure 4: Average scores of the 6 pieces (a). Style classification result of the 6 S1 (b).

6.2 Subjective style identification

In order to know if each trained model reflected the style observed in the training data, we conducted an another listening experiment for subjective style identification. 3 piano pieces, which were not included in the test pieces, were generated with both trained models (total 6 performances) and the participants listened to them to remember the style each model reflected. After that, the participants listened to the 12 S1 and S2 blind in a random order.

Figure 4b shows the style identification result of the 6 S1. The result shows that 5 out of 6 pieces were well identified by the listeners, and the average identification rate was 73.6%. The identification result of the 6 S2 was similar, and the average identification rate was 73.6%. Those results indicate that each trained model reflected the style observed in training data, and those styles were perceptually distinguishable by human listeners.

7. CONCLUSION

We introduced an automatic piano performance system called Polyhymnia that is able to learn and predict polyphonic expression, and interpret musical symbols automatically. Experimental evaluations on generated performances indicate that diverse performances of various compositions generated by the system had polyphonic expression and sounded expressively, and their performance styles were perceptually well distinguishable by human listeners.

We believe that modeling hierarchical structures of a given piece would improve a machine-rendered piano performance. By introducing additional model parameters controlled by users through an interface, Polyhymnia can be extended to an interactive music performance system.

8. REFERENCES

[1] L. Bottou. Stochastic gradient learning in neural networks. In *Proc. Neuro-Nimes*, Nimes, France, 1991. EC2.

[2] M. Hashida and et al. A new database describing deviation information of performance expressions. In *Proc. ISMIR*, pp. 489–494, 2008.

[3] D. Huron and et al. The avoidance of inner-voice entries: perceptual evidence and musical practice. *Music Perception*, 7(1):43–48, 1989.

[4] T. H. Kim and et al. Performance rendering for polyphonic piano music with a combination of probabilistic models for melody and harmony. In *Proc. SMC*, pp. 23–30, 2010.

[5] A. Kirke and et al. A survey of computer systems for expressive music performance. *ACM Comput. Surv.*, 42(1), 2009.

[6] J. Lafferty and et al. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pp. 282–289, 2001.

Multitouch Interface for Audio Mixing

Juan Pablo Carrascal
Universitat Pompeu Fabra
Music Technology Group
Carrer de Tanger, 122-140
08018 Barcelona, Spain
juanpcarrascal@gmail.com

Sergi Jordà
Universitat Pompeu Fabra
Music Technology Group
Carrer de Tanger, 122-140
08018 Barcelona, Spain
sergi.jorda@upf.edu

ABSTRACT

Audio mixing is the adjustment of relative volumes, panning and other parameters corresponding to different sound sources, in order to create a technically and aesthetically adequate sound sum. To do this, audio engineers employ "panpots" and faders, the standard controls in audio mixers. The design of such devices has remained practically unchanged for decades since their introduction. At the time, no usability studies seem to have been conducted on such devices, so one could question if they are really optimized for the task they are meant for.

This paper proposes a new set of controls that might be used to simplify and/or improve the performance of audio mixing tasks, taking into account the spatial characteristics of modern mixing technologies such as surround and 3D audio and making use of multitouch interface technologies. A preliminary usability test has shown promising results.

Keywords

audio mixing, multitouch, control surface, touchscreen

1. INTRODUCTION

Even though today we are listening to very high quality surround systems - both in theaters and in our homes - and we might be about to enter the 3D audio revolution [19, 14, 16], the interfaces that we are using for mixing audio do not seem to have changed much since their introduction. In electrical terms, mixing implies the adjustment of variable-resistance controls (faders or potentiometers), which are standard components for electronic devices. Thus, these are the controls which have been traditionally used in mixing desk design [3]. This potentiometer-based interface design has been used up until our days, even if it is not necessarily ergonomical or adequate. In software, no big redesign has been proposed either [6, 11], and the majority of recent multitouch interfaces are simple adaptations of the same control schemes [18].

Maybe it is time to use what has been learnt with HCI research and question a trend which has ruled mixing console design for decades. We propose an initial prototype in which we emphasize these fundamental features:

- It gives importance to the spatial quality of sound. It may make use of position in a 2D space as a fundamental parameter in the mixing process, but should also have the possibility to control the z-axis position, making it ready for 3D audio applications.
- The use of a *listening point* (LP)
- The use a metaphorical interface: instead of channel strips controls and output busses, there are *channels* located in a *stage*.
- The use of multitouch technologies

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. STATE OF THE ART

Peter Gibson [7] suggests a "Virtual Mixer", a virtual 3D space in which sound sources can be located in three physical axes that correspond to perceptual sound parameters. The snapshots shown seem visually useful and didactic, but somehow cluttered and thus not too practical from a HCI point of view or for professional applications.

In professional audio mixing, there are some interesting options such as the Mackie DXB[1], which extends a digital mixing console with a pair of single-touch screens.

Multitouch technology have shown interesting possibilites. A brilliant example is the JazzMutant Lemur[2] interface, and an increasing number of applications for mobile platforms. The trend in these cases is to emulate the layout of mixers [18], which is precisely what we want to avoid and challenge.

In last year's NIME, the Cuebert mixing board was presented [12]. It heavily integrates a multitouch interface to enhance a mixing console for musical theatre applications. However it still uses the same channel strip approach as traditional mixers.

Vincent Diamante [5] suggests an interface which has some common features with the one presented in this paper. We think it could be seen more as a data visualization tool than as a new interface for professional audio mixing. Also, Diamante's work does not consider 3D mixing technologies. It's important noticing that its features and the arguments in his justification can be taken as a confirmation that HCI design for audio mixing is worth to be explored.

3. DESIGN

It is important to remark that we are not proposing a *mixer* but a *control interface*. As it has happened with several musical applications since the introduction of MIDI, and in some novel audio mixing technologies such as Meyer's D-Mitri[3], the control interface is separated from the functional engine [9]. This would allow a low-processing-power unit (such as a tablet computer) to control a digital audio processing engine. Because of its flexibility, we chose Open Sound Control (OSC) [21] as the communication protocol between the interface and the sound processing unit.

[1] `http://www.mackie.com/products/digitalxbus/`
[2] `http://jazzmutant.com/`
[3] `http://www.meyersound.com/products/d-mitri/`

After some preliminary user tests with a paper prototype [17], a hi-fi prototype was developed to allow us to apply user tests and evaluate our ideas. The hardware platform we used was the Reactable [10], from which we only used its multitouch capabilities, without the fiducials. The prototype created allows the detection of up to four fingers, but further developments should be able to handle more. A finalized product could be implemented on a very lightweight and portable platform such as an Apple's iPad.

3.1 GUI

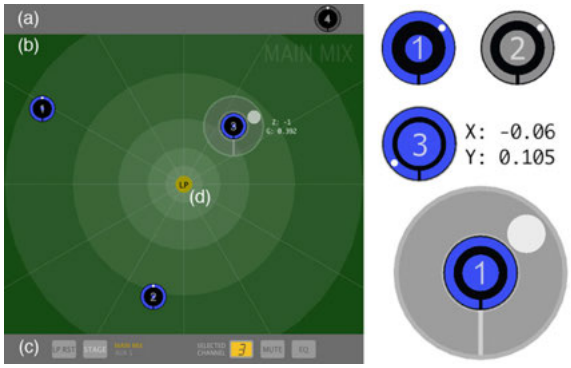


Figure 1: The GUI and the *channel*

The prototype has a simple yet flexible layout (Figure 1). There is a number (four in the prototype) of *channels* which can be dragged around the screen. There is an upper *inactive zone* (a), where all channels are initially located before being moved to the desired place. When channels are in this position, they remain muted and do not generate any control data. As soon as they are moved inside the *stage* (b), they become active. And just below the stage, there's a *control zone* (c) with buttons that control general functions: resetting the *Listening Point* (d) to it's default position, selecting between different stages, and displaying / muting / showing the equalizer for the currently selected channel.

3.1.1 The channel

We considered the channel to be the main element which should be changed from the traditional scheme. In standard mixers, it consists of a *channel strip* with lots of individual controls, mapped in a one-to-one basis to mixing parameters. Thus we took special care to create a control unit for it. We wanted to take advantage of the multitouch platform used, and we propose a versatile, multiparametric control scheme [8]. Of course, a control that can be dragged around a surface is going to have the inherent capability of controlling at least two parameters [4]. In our prototype, every channel (Figure 1, right) has these features:

- It can be dragged freely across x and y axes and Its position values are proportional to its panning and volume (in stereo mode) or to its surround (left-right and front-rear) panning (in surround mode).
- It has a gain control, by means of a surrounding "halo" with a marker that can be adjusted by moving it around the channel center.
- It has an internal circle whose diameter is proportional to the *z-axis* position parameter (for 3D audio environments). If the circle has a lower diameter, the channel is at a lower height, and viceversa.
- The value of the parameters is shown as they are adjusted.
- By watching the different channels in the stage it should be easy to spot and compare the values of their parameters at a glance.

Also, a fully functional, multitouch-enabled 4-band parametric equalizer interface is included for every channel.

3.1.2 The stage

Speaking in traditional mixing terms, the destination of the mix of a number of channels is called a *mixing bus* (the main mix, an auxiliary send, a recording bus, etc.). Electrically, this is just the cable that takes the electric signal from the summing circuit to the output connector [3]. We suggest the use of a metaphor for the bus in the interface. When mixing, we are going to locate sound sources in a sound space, or a *stage* (which represents the physical space, such as a studio or a live stage). Therefore, in our interface, each *stage* represents a possible destination for a sum of *channels*. For example, the main mix is one stage, an auxiliary mix for instrument monitoring could be another one, an auxiliary mix for a reverb effect could be yet another one and so on. Two stages were implemented in the prototype, a "Main Mix" and an "Aux 1" mix for adding a reverb effect. Every stage has a *listening point* (LP), and the panning of every channel present in the stage is determined by its position, relative to the position of the Listening Point. The Listening Point can be dragged around the stage with a two finger drag (to avoid accidental moves with a single finger). This way, it is easy to create custom mixes based on a reference mix without having to move every channel (Figure 2).

Figure 2: Two different mixes achieved by moving the Listening Point.

3.2 Functional Design

3.2.1 Software

We believed that a modular design process will make additional development or porting easier. Having that in mind, we chose Apple's *Quartz Composer* [2] for the development of our prototype. Quartz Composer (QC) is a visual, node-based programming language for graphical applications. It is released by Apple as part of the Xcode development tools.

Some of the patches (functional blocks used to create QC compositions) used in this project are not part of the basic QC distribution. These additional patches (Mansteri OSC sender[4], Kineme Structure Tools and Kineme Spooky Patch[5]) are, however, freely downloadable tools.

3.2.2 OSC address space

Currently, there are two main types of OSC messages generated by the interface, *stage* and *eq*, which are associated with stages and channels, respectively. This is the format of the *stage* messages (*muttmix* was chosen as the identifier for this project's OSC messages):

```
/muttmix/stage/N/ch/n A d E x y z g a
```

[4] `http://www.mansteri.com/software`
[5] `http://kineme.net/QuartzComposerPatches/`

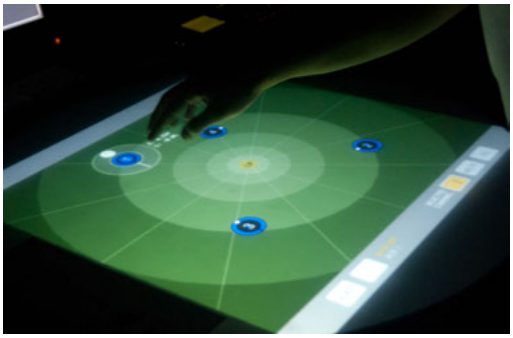


Figure 3: The prototype.

Where:

`N`: stage number
`n`: channel number
`A`: azimuth angle (float)
`d`: euclidean distance from LP (float)
`E`: elevation angle (float)
`x,y,z`: position - rectangular coordinates (float)
`g`: gain (float)
`a`: channel active (integer)

The position of the channel is sent as rectangular coordinates (x,y,z) as well as a 3D coordinate system compatible with the one used in Ambisonics [1] encoding and decoding. If the Listening Point is moved, every channel would send its updated position information, since their relative positions would change. This way the whole mix is readjusted to the new listening position.

The *eq* messages have this format:

```
/muttmix/eq/channel/n/b f g q G
```

Where:

`n`: channel number
`b`: band id (0 = low, 1 = mid-low, 2 = mid-high, 3 = high)
`f,g,q`: filter frequency, gain and Q (floats)
`G`: compensation gain (float) (Still not implemented)

3.2.3 *Communication with a mixer*

A Yamaha 01V96 [22] mixer, which has surround capabilities, was set up to be controlled by the proposed interface. For this particular setup, an additional tool was needed for translating and map the OSC messages generated by the multitouch interface into the appropriate MIDI continuous controllers understand by the 01V96. A custom Pure Data [15] patch was created for this purpose. The mixer was connected to a well calibrated surround monitoring system.

4. EVALUATION

A preliminary usability test was arranged, and two setups were made available in order to compare the performance of the users with both of them. The first setup is the same one described in the previous section. The other one involved using the controls of the 01V96 mixer directly. A four-track song was prepared, consisting of percussion, guitar, piano and voices. The basic working principles of both interfaces (01V96 and multitouch interface) were explained to all users (three men, three women, with ages ranging from 25 to 35; one of the men was a sound engineer, the rest had no previous experience in audio mixing). Users were asked to try to mix the song with the 01V96 mixer and with the prototype (in that order) trying to achieve certain specific spatial positioning of instruments. Though mixing has an inherent technical component, its aesthetic aspect is difficult to measure; there's not a precise "good" or "bad" way of doing it (specially taking into account that most of the users were non-experts). Because of this, users were asked to work as long as they want with both interfaces and try to achieve what they considered to be equally satisfactory results. The time required to finish the task was measured for both systems. A questionnaire with a Likert scale of 1 to 5 was applied afterwards to evaluate the interface, and users were asked to write their comments and observations. A video was shot during the tests, and users were also asked to sign a permission to use it in the context of this project.

4.1 Results

The times measured during the tests are shown in Table 1. The first column shows the times required by every user for completing the mix with the Yamaha 01V96, and the second column the times required with the proposed multitouch interface.

Table 1: Times used for mixing

User	Mixer	Multitouch
1	9:57	4:10
2	7:43	3:24
3	5:36	3:21
4	6:40	4:18
5	4:40	3:23
6	3:07	2:38
Average	5:33	3:25

5. DISCUSSION

An analysis of the comments written by the users showed preference for the multitouch interface over the mixer. Also, the multitouch interface seemed to be more time-efficient. For non-expert users, experimental results and user comments suggest that the multitouch interface was easier to learn. Interestingly, some users felt that it encourages creativity and playing more than the standard mixer. This might suggests further development for audio mixing education, especially in surround environments.

One user commented that the multitouch interface, more than the mixer, encouraged the use of both hands. At least two users were observed using both hands with the proposed interface and just one with the mixer. This might be due to the physical distribution of the mixer in the location, but it is an interesting point that should be further investigated.

Interestingly enough, the only expert user (first one in Table 1) had the longest time for the mixer and the second longest time for the multitouch interface. The user explained it saying that he took the time to make a very polished mix with both interfaces. This was not the case for other users who said to have felt a bit overwhelmed by some controls and in some cases opted for ignoring them.

We think that many considerations should be supported on expert users experience. Mixing is a task which involves technical and artistic components, both equally important. An expert should feel comfortable with the tool he or she is using in order to do a good job. Also, some studies suggest that the aesthetical features of an interface can affect its usability, and thus its perceived performance [20, 13]. So if a specific tool is deeply established in certain work context, a new one which offers not only a different set of functions, but also a different interface and aesthetic appeal

will be initially hard to accept. However, after showing the paper prototype and explaining the goals of this project to a well-known and experienced sound engineer, it was found that some of our concerns might have also appeared in the professional audio context. Projects such as the Reactable and others, many of them from the NIME conferences, have called for the attention of professional musicians towards new technologies. We believe that sound engineers might share the same interest.

6. CONCLUSIONS AND FUTURE WORK

We presented the prototype for a novel multitouch mixing control interface, supported with a preliminary evaluation by means of user tests and questionnaires.

The emphasis on spatial control, relating mixing parameters with physical position of the input channels is one of the strong points of the interface. The "channel+stage" approach seemed intuitive for novice users, and offered a true metaphor-based interface as opposed to traditional mixers.

The literature and the test results suggest great possibilities for interfaces like the one suggested. The biggest drawback of the proposed system, as in any touchscreen, is the lack of tactile feedback. Traditional mixing consoles, with lots of physical controls, have a great advantage in this aspect. Hopefully a fully developed prototype, a better implementation, and good demonstration strategies would make the proposed approach more competitive.

6.1 Future Work

The shape and size of the Reactable are not ideal for the context of professional audio mixing, so it would be desirable to port our prototype to a multitouch platform with a smaller size, a higher graphical resolution, and a rectangular surface, such as a tablet computer.

Some of the findings gathered during the paper prototype stage have not been implemented yet, and some of them, specially the ones coming from expert users, are crucial. Additional functionalities, such as the possibility to control dynamic processors and external plug-ins, would add value to the package. In general, a competitive prototype should allow the user to do anything he could do with a hardware mixer, in order to perform a fair comparison. This was out of the scope of this project, but would be highly desirable for a final implementation.

Last but not least, more statistically significant tests are yet to be done. These should be done after further refining the prototype, and involving more expert users. However, so far, the results are promising.

A comparison with the software mixer of a digital audio workstation such as Pro Tools was planned but not done due to technical and time constraints. It would be good to include this option in further tests.

7. REFERENCES

[1] T. W. Abhayapala and D. B. Ward. Theory and design of high order sound field microphones using spherical microphone array. In *Proceedings of ICASSP*, 2002.

[2] Apple. *Introduction to Quartz Composer User Guide.* Apple, Inc.

[3] G. Balou. *Handbook for Sound Engineers. The new audio cyclopedia.* Howard W. Sams and Company, 1 edition, 1987.

[4] W. Buxton. Lexical and pragmatic considerations of input structures. *SIGGRAPH Comput. Graph.*, 17:31–37, January 1983.

[5] V. Diamante. Awol: Control surfaces and visualization for surround creation. Technical report, University of Southern California, Interactive Media Division., 2007.

[6] M. Duignan, J. Noble, P. Barr, and R. Biddle. Metaphors for electronic music production in reason and live. In *6th Asia-Pacific Conference on Computer-Human Interaction*, 2004.

[7] D. P. Gibson. *The Art of Mixing.* ArtistPro Press, 1997.

[8] A. Hunt, M. M. Wanderley, and R. Kirk. Towards a model for instrumental mapping in expert musical interaction. In *San Francisco: International Computer Music Conference*, 2000.

[9] S. Jordà. New musical interfaces and new music-making paradigms. In *Proceedings of the 2001 conference on New interfaces for musical expression*, NIME '01, pages 1–5, Singapore, Singapore, 2001. National University of Singapore.

[10] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, pages 139–146, New York, NY, USA, 2007. ACM.

[11] G. Levin. Painterly interfaces for audiovisual performance. Master's thesis, Massachusetts Institute of Technology, 1994.

[12] N. Liebman, M. Nagara, J. Spiewla, and E. Zolkosky. Cuebert: A new mixing board concept for musical theatre. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, 2010.

[13] S. R. McDougall and I. Reppa. Why do i like it? the relationships between icon characteristics, user performance and aesthetic appeal. In *Proceedings of the Human Factors and Ergonomics Society 52th annual meeting.*, 2008.

[14] N. Peters, T. Matthews, J. Braasch, and S. McAdams. Spatial sound rendering in max/msp with vimic. In *Proceedings of the 2008 International Computer Music Conference*, 2008.

[15] M. Puckette. Pure data. In *Proceedings of the International Computer Music Conference, (ICMC).*, 1996.

[16] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. In *The journal of the Audio Engineering Society*, volume 45, 1997.

[17] M. Rettig. Prototyping for tiny finger. *Communications of the ACM*, 37(4):21–27, 1994.

[18] C. Roberts. Multi-touch, consumers and developers. Technical report, Media Arts and Technology Program - University of California, 2008.

[19] J. K. Thompson. The allobrain: An interactive, stereographic, 3d audio, immersive virtual world. In *International Journal of Human-Computer Studies*, volume 67, 2009.

[20] N. Tractinsky, A. S. Katz, and D. Ikar. What is beautiful is usable. *Interacting with Computers*, 13(2):127 – 145, 2000.

[21] M. Wright and A. Freed. Open sound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the International Computer Music Conference*, 1997.

[22] Yamaha. *Manual of the Yamaha 01V96 digital mixing console.* Yamaha Corporation.

Cognitive Architecture in Mobile Music Interactions

Nate Derbinsky
Computer Science & Engineering Division
University of Michigan
2260 Hayward Ave
Ann Arbor, MI 48109-2121
nlderbin@umich.edu

Georg Essl
Electrical Engineering & Computer Science and Music
University of Michigan
2260 Hayward Ave
Ann Arbor, MI 48109-2121
gessl@eecs.umich.edu

ABSTRACT

This paper explores how a general cognitive architecture can pragmatically facilitate the development and exploration of interactive music interfaces on a mobile platform. To this end we integrated the Soar cognitive architecture into the mobile music meta-environment urMus. We develop and demonstrate four artificial agents which use diverse learning mechanisms within two mobile music interfaces. We also include details of the computational performance of these agents, evincing that the architecture can support real-time interactivity on modern commodity hardware.

Keywords

mobile music, machine learning, cognitive architecture

1. INTRODUCTION

How can contemporary work in machine learning and cognitive architectures be used in mobile music interactions? Here we integrate a contemporary cognitive architecture with an emerging mobile music environment and show the pragmatic use of various learning stategies in this context.

The introduction of interactive music-making techniques has shown some impressive outcomes. Fiebrink *et al* [6] have demonstrated that supervised machine learning can be used to define interactive gesture-based music applications on laptops. However the introduction of comparable ideas to mobile music interaction is lacking. Current mobile smart devices are different from laptops in the kinds of interactions that are natural to perform on them and the kinds of sensors that are available on them. For example hand gestures are a rather natural mode of engagement with a mobile device, whereas accelerometer-based interactions on laptops are possible but have a distinctly different flavor. Further mobile smart devices are available to a larger demographic than laptops suggesting the need to support them as primary computational platforms [4].

In this paper, we explore interactive learning and musical expression on mobile devices. In contrast to prior work that applied specialized machine learning algorithms, we use a cognitive architecture, a system that efficiently and generally integrates multiple learning and memory modules for use across numerous tasks.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. COGNITIVE ARCHITECTURE

The central goal of artificial intelligence is the development and understanding of intelligent agents, autonomous entities that observe and act within an environment, applying human-level reasoning capabilities to achieve their goals. Many researchers in the field, however, do not work directly at the level of generally intelligent agents, but instead strive to understand one or more sub-problems in specific contexts, such as machine learning, the study and development of algorithms to find patterns in empirical data; planning and reasoning, especially under uncertainty; and computational processing and generation of natural language.

By contrast, research into cognitive architecture aims to develop and understand human-level intelligence across a diverse set of tasks and domains [8]. A cognitive architecture is a specification of those aspects of cognition that remain constant throughout the lifetime of an agent. These fixed components include short- and long-term memories of the agent's beliefs, goals, and experience; the representation of elements contained within these knowledge stores; functional processes that apply agent knowledge to produce behavior; and learning mechanisms that adapt agent knowledge over time. Cognitive architecture applies a systems-level approach to artificial intelligence research, investigating how the integration of numerous computational mechanisms supports complex and adaptive behavior.

Diverse cognitive architectures have been developed over the last forty years, but nearly all specific cognitive architecture research efforts strive towards at least one of the following three goals: (1) biological plausibility, (2) psychological plausibility, and (3) agent functionality. For instance, systems such as Leabra [10] attempt to computationally explore how intelligence arises from circuits of neurons and how architectural mechanisms and processes correspond to neurobiological data regarding brain regions and topological connectivity. By contrast, systems such as EPIC [9] and ACT-R [1] are typically applied at a layer above biological mechanisms and attempt to capture and model details of human performance, such as behavioral timing and memory recall errors, in a wide range of cognitive tasks. Finally, architectures like Soar [7] strive to understand how human-level intelligence arises from computational architecture and are typically applied as an effective path to building broadly capable artificial agents.

There are two primary appeals of considering cognitive architectures for music performance. The first involves quality of interaction with a learning system. The response of an interaction may involve familiar characteristics, such as remembering or forgetting musical phrases. Agent functionality can offer the appearance of such cognitive function in a way that a performer can potentially relate to, hence making the machine learning process itself more intuitive. The second reason is one of development pragmatism. Typical

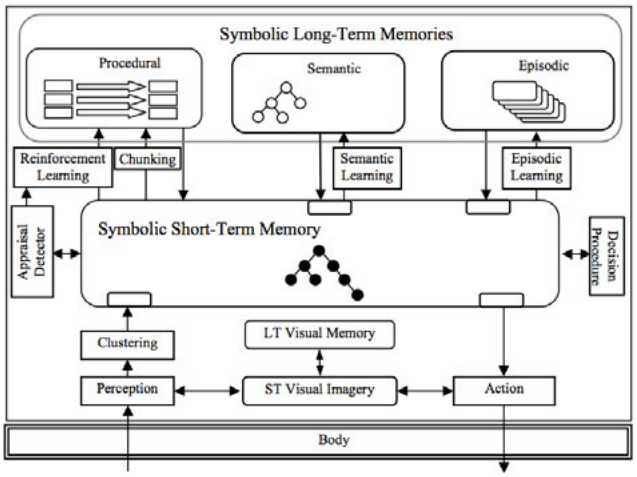


Figure 1: The structure of the Soar cognitive architecture.

applications of machine learning techniques involve specialized algorithms and thus there is significant burden on developers to implement one or more algorithms, tune them for a particular task, including any integration issues that arise, and finally employ optimization techniques such that the algorithms scale to complex problems, an especially difficult challenge on mobile platforms. These burdens are lessened with the application of cognitive architecture wherein the locus of development is declaring agent knowledge and goals.

In addition to research goals, individual cognitive architectures deviate along numerous dimensions. We considered two metrics in selection of the candidate architecture. First, to explore complex musical expression, we sought an architecture that could process over, as well as reason and learn about, diverse information sources, including temporal sequences of musical notes and declarative rules of music composition. Second, to support interactive mobile tools, we we sought an architecture that could bring these knowledge sources to bear while maintaining real-time decision making. Thus, we applied and evaluated Soar [7], a functionally driven cognitive architecture.

3. SOAR

Soar is a cognitive architecture that has been used extensively for developing artificial intelligence applications and modeling human cognition. One of Soar's main strengths has been its ability to efficiently represent and bring to bear large bodies of symbolic knowledge to solve diverse problems using a variety of methods [7]. Soar supports a variety of programming languages (such as C++, Java, and Python) on all major operating systems (including Windows, Mac OS, Linux, and iOS) and has been interfaced in diverse execution environments, including game systems and robotics simulation and hardware platforms.

Figure 1 shows the structure of Soar. At the center is a symbolic working memory, represented as a graph, that captures the agent's current state. Perception from the world, such as sights, sounds, or contact, delivers symbolic structures to working memory. The long-term memories retrieve information based on the contents of this working memory and add, delete, or modify these structures. The procedural memory, encoded as if-then rules, captures the agent's knowledge of when and how to perform actions, both internal, such as deliberately querying other long-term memories, and external, such as the production of sound through speakers or control of robotic actuators. This knowledge can be tuned over time by the integrated reinforcement learning [11] mechanism, which adjusts the selection of actions in an attempt to maximize receipt of reward. The semantic long-term memory encodes general facts about the world, which may be pre-loaded from existing knowledge bases, while episodic memory incrementally builds an autobiographical history of agent experience. As evident in Figure 1, Soar has additional memories and processing modules; however, they are not evaluated in this paper, and are not discussed further.

Processing in Soar is decomposed into a sequence of decisions. The basic *decision cycle* is to process input, fire rules that match, make a decision, fire rules that apply the decision, and then process output commands and retrievals from long-term memory. The time to execute this processing cycle determines reactivity and so our evaluation will include (1) the number of decisions that were made to complete the task, (2) the average amount of time to execute each cycle, (3) the maximum amount of time required for any cycle, and (4) the total CPU time consumed by the Soar agent completing the task.

4. INTEGRATING SOAR IN URMUS

UrMus [5] is set up to provide a flexible system to receive input and organize visual and other content in response. The main organizing element for multi-touch input as well as visual output are regions. They allow maximum flexibility in designing interactions. Hence it felt natural to associate an instance of Soar with regions. Each region can have an instance of Soar attached, hence one can have one or more agents for each input element and agents for each movable visual element. Other media elements can be realized by having region-based events instantiate Flowboxes (elements of UrMus's dataflow engine). This means it is easy to have objects that independently navigate, say, the screen-space to have independent cognitive models running.

The Soar kernel is implemented in C++ and we have integrated the architecture with urMus via a minimal Lua interface that allows urMus to supply perception to the agent, including touch events from the user, read actions decided upon by the agent, such as producing a note, and execute arbitrary commands, such as to control the agent's run state and illicit debugging and computational performance information.

To see how this works, let us consider the following simple example code. As said, each region can have a Soar agent attached to it. In order to do anything meaningful, this agent will need a rule set to be loaded:

```
r = Region()
r:SoarLoadRules("simon-rl", "soar")
```

In order to learn, a percept (in form of a symbolic constant) is created in Soar's input data structure and a learning step is executed. In this case a notion of time is learned that is derived from a user interaction. The input is deleted when done:

```
timeWme = r:SoarCreateConstant(0,
                    "time", clickcount)
r:SoarExec("step "..delayDecisions)
r:SoarDelete(timeWme)
```

In order to generate output after learning, one can read the output from Soar's output data structure:

```
taskWme = r:SoarCreateConstant(0,
                    "task", "generate")
name, params =  backdrop:SoarGetOutput()
```

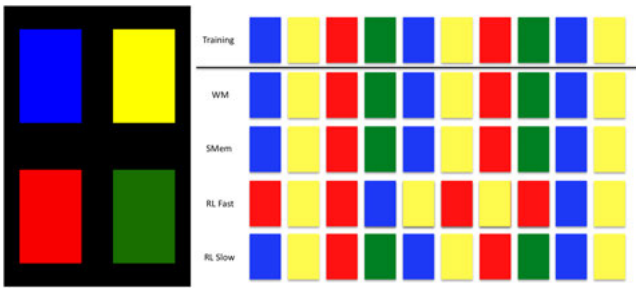


Figure 2: Simon demonstration with example input/output sequences.

```
result = params.output
r:SoarSetOutputStatus(1)
r:SoarDelete(taskWme)
```

Here `output` is the entry of interest. There are also other functions that help control a Soar agent, such as halting the agent with the `r:SoarFinish()` function and reinitializing the agent anew with the `r:SoarInit()` command.

5. DEMONSTRATIONS

We now present two musical demonstrations implemented in the UrMus environment [5] using Soar 9 [7] and deployed to the iOS platform on iPhone, iPad, and iPod Touch hardware. These tools are not intended to represent state-of-the-art music interface design, but instead demonstrate how cognitive architecture can facilitate the development of interactive, novel musical tools on mobile platforms.

5.1 Simon

Our first demonstration was to implement Soar agents playing Simon, a game of memory skill [2] illustrated in Figure 2. In this demonstration, the user inputs a sequence, selecting from four colored buttons, each of which emits a different musical tone. The button presses are provided to the Soar agent sequentially in real-time and after input is complete, the Soar agent generates an arbitrary length of musical response, based upon its knowledge and learning of the input stream. The focus of this demonstration is to explore how utilizing different memory models, both short- and long-term, can affect development of musical interfaces. For clarity, the Soar cognitive architecture is held constant for all demonstration agents below, whereas the agent's initial procedural knowledge, encoded as if-then rules, is what is altered such as to distinguish each agent's behavior.

In this task, our first evaluation metric was accuracy of the agent's response - to what extent does the agent reproduce the input sequence? While a challenge for humans, especially over long input sequences, one could imagine basic algorithms that could store and generate a fixed length button string. Thus, our second, more interesting evaluation dealt with output generativity, or the degree to which the agent could produce novel output, while adhering to the "spirit" of the input sequence.

The first agent we developed appended new button inputs to an endless linked list within the agent's working memory. As expected, the result of this agent was perfect input reproduction, at the cost of learning no generalization over the input. After receiving input and producing an output sequence of 10 buttons, the agent required 415 decision cycles, averaging 0.053 milliseconds per decision with a maximum of 1 millisecond for a decision, for a total of 0.022 CPU seconds.

The next agent applied an instance-based learning approach, storing all inputs to its semantic memory. We encoded each note as a simple (previous note, next note) pair and thus generation of new musical sequences simply retrieved the "next" note conditioned upon the last note played. Figure 3 captures the state of the agent's semantic memory after it had heard the sequence of three notes: A, B, C (where one/two refer to previous/next and *nil* refers to the beginning of the sequence). For very short, distinct button sequences (fewer than four buttons, no repetition), it is possible for the agent to perfectly reproduce the input. However, the common case is for ambiguities to arise in the input sequence, which are disambiguated via a recency bias in memory retrieval [3]. This agent required 278 decisions at an average of 0.086 milliseconds per decision, requiring a maximum of 1 millisecond in a decision, for a total of 0.024 seconds of CPU time.

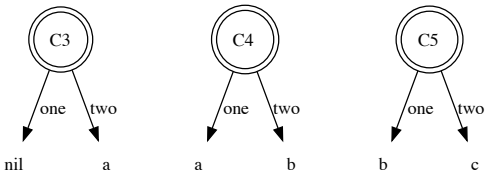


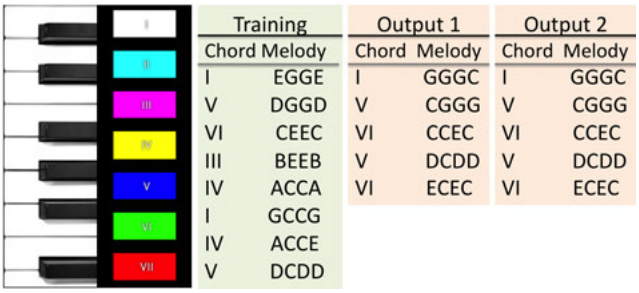
Figure 3: Simon semantic memory representation.

Our final agent applied Soar's reinforcement learning mechanism to this memory task. During the user's input, the agent "practiced" the known sequence, given the same state representation as the semantic memory agent, self-rewarding for reproducing the correct input. For instance, the following rule captures some practiced knowledge after the agent heard the sequence of three notes: A, B, C.

```
If
   no previous note was heard AND
   the agent is considering producing note C
Then
   the expected value of this decision is -5
```

The value at the conclusion of the rule was updated over time based upon experience and practice. The amount of practice was directly proportional to the amount of time between user inputs. As a result, the generated musical output was affected by the sequence of buttons the user pressed, the time taken to input the sequence, and the probabilistic application of the agent's learned music generation policy. We found that given enough time between button presses (about 1 second), the agent would frequently reproduce most of the initial input sequence, while occasionally deviating to produce novel, probabilistically derived subsequences. We provided the same input sequence to the agent twice, varying only the amount of time between button presses (such as to change available practice time). The agent given little practice ran for 827 decisions, averaging 0.145 milliseconds per decision with a maximum of 1 millisecond for a decision, totally 0.120 seconds of CPU time. The agent with more time ran for 1133 decisions, averaging 0.176 milliseconds per decision with a maximum of 1 millisecond, totalling 0.199 seconds of CPU time.

The agent performances in the Simon demonstration are summarized in Figure 2 for a particular input sequence ("Training). The working memory ("WM") agent perfectly reproduces the input, as it cannot generalize. The semantic memory agent ("SMem"), with a limited instance-based representation, primarily reproduces the input (as seen in Figure 2), with small deviations when the next note is not uniquely determined by the previous note. The reinforcement learning agent is shown with two amounts of practice



Training		Output 1		Output 2	
Chord	Melody	Chord	Melody	Chord	Melody
I	EGGE	I	GGGC	I	GGGC
V	DGGD	V	CGGG	V	CGGG
VI	CEEC	VI	CCEC	VI	CCEC
III	BEEB	V	DCDD	V	DCDD
IV	ACCA	VI	ECEC	VI	ECEC
I	GCCG				
IV	ACCE				
V	DCDD				

Figure 4: The interface of the music generation urMus/Soar implementation (left). Training and two generated results using the reinforcement learning.

time ("RL Fast" and "RL Slow"), illustrating probabilistic differences from the training sequence. If the learning time is fast the sequence is more likely to deviate from transitions seen in learning, whereas longer learning will reinforce transitions that are seen frequently hence lead to sequences that more closely resemble the original. However, these models are probabilistic and hence do not guarantee reproduction. For musical purposes this is interesting because it means that learning the rules of production are reinforced but variation is retained.

5.2 Mobile Music Generation and Interaction

Our next demonstration tasked the agent with generating simple musical scores after perceiving a sequence of chords accompanying musical notes (see interface in Figure 4). To simplify the quantization problem, the time scale of the input was fixed (one chord per 4 notes). Once again, our evaluation considered both the accuracy of music reproduction, as well as novelty of musical generation.

For this demonstration, we extended our reinforcement learning agent from the Simon task such as to simultaneously learn chord sequencing, note sequencing, and note-chord association. The following are two representative rules learned by the agent:

```
If
   the current chord is C-E-G AND
   the previous note played was G AND
   the agent is considering producing note C
Then
   the expected value of this action is -8

If
   the previous chord was C-E-G AND
   the agent is considering chord E-G-B
Then
   the expected value of this decision is -8
```

The agent's practice time (limited by real-time interaction with the user) was split between learning chord sequencing, note sequencing, and note-chord association. We found that given sufficient "practice" time (about 100 decisions between notes), the Soar agent was able to associate notes with chords and produce similar chord sequences as the input, though note sequencing was unimpressive in reproduction, nor generative quality. In this task, our agent required 360 decisions, averaging 0.217 milliseconds per decision and requiring a maximum of 1 millisecond for a decision and a total of .118 seconds of CPU time.

An example training set of eight chords with four note monophonic melodies each and two generated outputs of five chords with four note melodies each are shown in Figure 4. Each run of Soar will generate a new output and note that the adherence to learned rules is not yet very strict. Notes that do not strictly belong to the underlying chord are played. The training set contains one such exception. With repeated training the melodies become more reflective of the input. This model can be run offline or interactively in a call and response scheme. The user plays a chord and four notes and the system will generate the same based on what it has learned so far. Over time the call and response duet locks into a more stable style as the learning algorithm reinforces the observed rules of the played call melodies.

6. CONCLUSIONS

In this paper we showed how the integration of a cognitive architecture and a mobile music platform can lead to novel forms of interactive music expression. We developed two systems that demonstrate how interactive musical tools can benefit from complex, integrated applications of machine learning algorithms (such as reinforcement learning) and instance-based learning (such as declarative retrievals from semantic memory). We also showed that the Soar cognitive architecture is sufficiently efficient to support interactive musical tools on a mobile platform. These demonstrations, however, do not begin to explore the space of possibilities a cognitive architecture offers to the development of novel mobile musical tools.

7. REFERENCES

[1] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An Integrated Theory of the Mind. *Psychological Review*, 111:1036–1060, 2004.

[2] R. H. Baer and H. J. Morrison. Microcomputer Controlled Game - US Patent 4,207,087, 1980.

[3] N. Derbinsky, J. E. Laird, and B. Smith. Towards Efficiently Supporting Large Symbolic Declarative Memories. In *Proceedings of the 10th International Conference on Cognitive Modeling*, 2010.

[4] G. Essl. Mobile Phones as Programming Platforms. *Proceedings of the First International Workshop on Programming Methods for Mobile and Pervasive Systems*, 2010.

[5] G. Essl. UrMus-an environment for mobile instrument design and performance. *Proceedings of the International Computer Music Conference*, 2010.

[6] R. Fiebrink. *Real-time human interaction with supervised learning algorithms for music composition and performance*. Disseration, Princeton University, 2011.

[7] J. E. Laird. Extending the Soar Cognitive Architecture. In *Proceedings of the First Conference on Artificial General Intelligence*, Memphis, TN, 2008. IOS Press.

[8] P. Langley, J. E. Laird, and E. Rogers. Cognitive Architectures: Research Issues and Challenges. *Cognitive Systems Research*, 10(2):141–160, 2009.

[9] D. E. Meyer and D. Kieras. A Computational Theory of Executive Control Processes and Human Multiple-Task Performance. Part 1: Basic Mechanisms. *Psychological Review*, 1997.

[10] R. C. O'Reilly and Y. Munakata. *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. MIT Press, Cambridge, MA, 2000.

[11] R. S. Sutton and A. J. Barton. *Reinforcement Learning: An Introduction*. 1998.

The Self-Supervising Machine

Benjamin D. Smith
University of Illinois at Urbana-Champaign
School of Music
Urbana, Illinois
bdsmith3@illinois.edu

Guy E. Garnett
University of Illinois at Urbana-Champaign
eDream, Illinois Informatics Institute
Urbana, Illinois
garnett@illinois.edu

ABSTRACT

Supervised machine learning enables complex many-to-many mappings and control schemes needed in interactive performance systems. One of the persistent problems in these applications is generating, identifying and choosing input output pairings for training. This poses problems of scope (limiting the realm of potential control inputs), effort (requiring significant pre-performance training time), and cognitive load (forcing the performer to learn and remember the control areas). We discuss the creation and implementation of an automatic "supervisor," using unsupervised machine learning algorithms to train a supervised neural network on the fly. This hierarchical arrangement enables network training in real time based on the musical or gestural control inputs employed in a performance, aiming at freeing the performer to operate in a creative, intuitive realm, making the machine control transparent and automatic. Three implementations of this *self supervised* model driven by iPod, iPad, and acoustic violin are described.

Keywords

NIME, machine learning, interactive computer music, machine listening, improvisation, adaptive resonance theory

1. INTRODUCTION

Machine learning (ML) continues to gain increasing application in the performing arts as the problems of interactive system control in live performance become more and more approachable and better understood [7]. The promise of the transparent union of live performer and complex multimedia performance is alluring, and with the development of on-line ML algorithms and the computing power required to run them in real time, such performances are rapidly becoming reality. Yet the extensive pre-performance training required of most musical ML applications poses a particular problem to the improvising musician who wishes to privilege spontaneous musical creativity during a performance.

We describe herein the design of a unique self-supervised system that employs unsupervised learning algorithms, specifically Adaptive Resonance Theory (ART), to automatically parse input music and gesture streams, locate significant feature areas, and train many-to-many mappings in real time. The result is an interactive multi-media system that generates mappings uniquely for each performance, extracting the particulars of a given input stream and creating a control space that produces rapid, tightly coupled responses.

2. MOTIVATION

The applicability of ML methods to problems in interactive musical performance is evidenced by the number and variety of applications and cases (see for example [4, 14, 16]). Recent systems, such as the work of Fiebrink et al. [7], focus both on real-time training, in order to better match the musician's work process, as well as the use of ML to discover new musical expressions. Rather than attempt to exactly duplicate preconceived mappings they encourage the exploration of unexpected results stemming from active training during a performance.

However, supervised ML implementations conventionally require that the musician define both their input material (i.e. what the system will learn to identify) as well as the desired outputs (the intended results in an interactive performance system) in advance of their use during a performance. Training the computer to produce desired outputs for given inputs serves to effectively build a complex computer music instrument driven by dynamic gestural controllers or acoustic instruments. This may be accomplished transparently during a performance [7], but requires extensive awareness and expertise on the part of the performer.

For example, consider a musician who, during an improvisation, trains the system to recognize two distinct melodic patterns, tying these to two different system outputs. As the piece progresses, the musician must anticipate their own movement to new melodic areas, retraining the system at each point. Failing this, the performer's connection to the computer becomes challenged, as the content of the music, i.e. the relationships between notes, events, and phrases, moves away from the domain that the system was trained for. This will be especially apparent in systems designed for discrete classification, i.e. where the system only produces outputs when a known input is observed, but is also problematic with interpolating systems due to the input's movement away from the known feature domain. This can be partially obviated by providing a sufficiently broad range of inputs to train on, but this requires a very substantial effort in defining or predefining suitable training sets.

Thus these systems can be prohibitive for an improvising musician who wishes to privilege spontaneity and creativity in the moment of the performance. Additionally, existing systems typically treat all categories equally, missing the musical interest resident in the inter-input relationships. We mitigate this dependence on predetermination of scope, by designing a system that automatically identifies movement to new areas of the input set (i.e. melody, in this example), and creates mappings to account for these per-

ceptive, musical developments.

These interactively determined relationships are better captured by unsupervised, on-line ML models, though the latter have seen virtually no application in interactive performance to date. These models allow the algorithm to discover classifications and find groupings and patterns across inputs based on relationships inherent in the data, rather than training on preconceived knowledge of the inputs (such as [1, 6, 10] applied to problems in Music Information Retrieval). Effectively, the computer is allowed to build its own interpretation of the musical work, listening in a fashion analogous to the human listener [3]. The primary problem posed by unsupervised methods is the potential for input to be categorized in ways unanticipated by the performer (however, this is also a possibility when playing with other humans). Inherently in "unsupervised" algorithms, the inputs are automatically parsed and categorized without human oversight or labels.

The capability of unsupervised learning to analyze musical material is shown by Gjerdingen [8] and Piat [12], who employ ART models to produce automatic classifications. The former found that the machine could automatically learn to identify formal changes in early Mozart compositions, discovering relationships without human supervision. Piat used a similar system to train a computer to hear the relative difference between "consonant" and "dissonant" music, comparable to human test subjects with surprising fidelity. However, neither of these projects ran in real-time.

Our ART implementation is faithful to [2], although our work appears to be the first real-time, performance oriented application of ART in music.

3. DESIGN

The design of our system consists of two primary modules, a *supervisor* component and the mapping network. The supervisor has an ART network at its core, itself running without supervision, examining the input feature stream for pitch-focal areas (in the case of musical input), and producing categorizations. The mapping is accomplished through a MLP network, enabling non-linear translation of inputs to outputs. The outputs must be defined in advance, as in [4, 7, 14, 16].

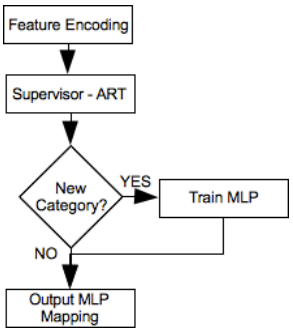


Figure 1: System ML interaction.

The connection between the two networks is unidirectional, with the supervisor acting as a trigger to retrain the MLP. Given a stream of feature data the ART will classify and group the inputs, creating categories of varying size that parse the total feature space. When a category is created, and the overall system decides it is a significant category, it is tied to a predetermined output and a training session commences. In the current implementations the realm of possible outputs is defined by a data set created in advance of operation. This is generated effectively through a pre-performance *improvisation* wherein the composer/performer chooses the domain of system outputs and orders them sequentially. As the performance unfolds the output set is traversed in order, creating mappings to the live inputs. The training session runs in the system's free time between input presentations, allowing the performer to continue uninterrupted by the training.

The system as described is currently embedded in three distinct implementations, driving audio synthesis and video animation with mobile touch devices (Apple iPod and iPad), as well as acoustic instruments (a violin).

3.1 Supervisor: ART

The supervisor models both a short-term memory (retaining the last several seconds of input) and long-term memory (grouping, relating, and retaining inputs for later recall) based on contemporary theories of human audition and perception [11, 15]. This is accomplished as two distinct components: a feature encoding module (the short-term memory, STM) and the unsupervised ART implementation (the long-term memory). The aim is to provide the computer with a method of parsing the feature inputs that mimics a human's abilities, such that a distinctive, to a human, change in the inputs (and in the originating music or gestures) is recognizable by the machine.

3.1.1 Short-Term Memory

The feature encoding can be accomplished in a number of ways, dependent on the nature of the input data. Our implementations employ two distinct models. The first is a simple scaling and grouping of independent values representing significant components of the input data. In the case of the mobile touch implementations this involves tracking the touch position(s), touch velocity (delta between touch samples), and touch curvature (degree of change in direction between velocity samples proportional to distance traveled), and transforming these values into a vector where each element (x) obeys: $0 \leq x \leq 1$.

The same model is similarly employed for part of the analysis of the acoustic violin input, creating a feature vector from spectral centroid ("center of mass" of the spectrum, or perceptual "brightness" [13]), spectral noisiness (how tone-like or noise-like the sound is), and register (average pitch over a two second window).

Spatial encoding [5, 8] comprises the other primary STM model, enabling the transformation of melodic and pitch sequences into feature data that the ART can process. A simple neural network with attenuated feedback forms its core, with one node for each unique token in the potential input set (this model is commonly used in natural language processing where each character in an alphabet is used as a token). When a token is presented to the network the corresponding node is fully activated and the network produces an output (x). This output is in turn fed back into the network, attenuated by a small amount (α) (typically set to retain five to nine inputs in the network [9]):

$$x_t = \alpha x_{t-1} \tag{1}$$

Thus the occurrence of each token in time is transformed into a vector, where each element indicates relatively how recently that token appeared. The vector can then be understood as a spatial representation of a point in the sequence, creating a position and movement within the token space. For musical input, pitch classes and interval classes (based on the twelve-tone equal-tempered tuning system) become ready token sets, creating feature vectors that depict melodic and harmonic movement and allow the detection of motivic and pitch-class set relationships (see [8]).

3.1.2 Long-Term Memory

Once the STM is created, the resulting vectors are fed to the ART module for classification. The ART is a competitive neural-network using unsupervised training, creating feature categories based on an ordered sequence of input vectors. That the input is ordered is significant, as different orderings of the same data will produce divergent classifications. While usually considered a limitation, this is an asset in music parsing where the order is carefully contrived by the artist and is important—and usually specific—to the particular musical work. Just as a human listener relates later melodic development (such as a sonata-form development section, or recapitulation) to earlier auditions (i.e. the exposition), so does the ART algorithm.

When presented with a new input vector ($\boldsymbol{I}$) the ART algorithm first obtains a resonance measure (T) through the comparison of each known category ($\boldsymbol{w}$) with the new input (Eq. 2).

$$T_j(\mathbf{I}) = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\gamma + |\mathbf{w}_j|} \tag{2}$$

For a given input ($\boldsymbol{I}$) the resonance measure is calculated with choice function (T), comparing the input with the adaptive weights ($\boldsymbol{w}$) of each category (j). A choice parameter (γ) affects the matching of inputs to the closest subset category, and is typically set close to 0 to achieve this. The "fuzzy AND" operator $\wedge$ is defined by

$$(\mathbf{x} \wedge \mathbf{y}) = min(x_i, y_i) \tag{3}$$

and the norm $|\bullet|$ is the L1 norm

$$|\mathbf{x}| = \sum_{i=1} |x_i| \tag{4}$$

Before learning ensues the strongest resonating node must pass a "vigilance" test, to ensure it remains within a preset limit (Eq. 5). If the node's size is acceptable then it is selected and allowed to learn based on the input. On the other hand if by incorporating the new input the category size (in feature space) would increase beyond this limit (the "vigilance" parameter, p, in Eq. 5), then this node is rejected for this iteration and the next most resonant node is considered.

$$\frac{|\mathbf{I} \wedge \mathbf{w_j}|}{|\mathbf{I}|} < p \tag{5}$$

The rejection of all existing category nodes results in the creation and training of a new category node. The details of the ART algorithm we employ are described at greater length by Carpenter et al. [2].

The other control parameter of significance is the "learning rate" of the ART network. This parameter allows the network to both train new inputs immediately and still adapt slowly, retaining the identity of older categories. Setting the learning rate high causes categories to expand and fully incorporate new inputs while setting it low causes the categories to adjust slowly, settling into an average area of the feature space. The implementations below set the learning rate near 1, allowing identified categories to adapt to new inputs immediately, ensuring reproducible classifications (setting the learning rate low can cause subsequent presentations of the same inputs to be classified differently, dependent on category expansion rates).

3.2 Mapping: MLP

The MLP is a feedforward neural-network, consisting of multiple layers of nodes fully connected in a directed graph, which maps sets of input data onto sets of output data. The input nodes have a simple linear activation function but the nodes of the hidden internal layers have non-linear activation functions (typically, and in our case, these are sigmoids). Backpropagation is used to train the network, repeating iteratively over the course of a performance session.

Training of the MLP occurs frequently but unobtrusively to the user as new data is created during a performance. The ART module generates paired sets of inputs and outputs, which are updated based on musical (or gestural) developments as the work unfolds. For every updated training set a training period is initiated wherein the inputs and outputs are presented to the backpropagation algorithm iteratively until an error function falls below a given threshold or a safety time-out is reached. For the error function we employ a simple distance measure of the amount of correction the network undergoes for each new input set (i.e. an average of how much each node's parameters were changed during the training iteration). When the error condition is met (typically, the average change is below 10^{-5}) the training session ceases. The safety time-out is triggered when the error condition fails to move below a higher threshold (typically 0.01) within 20,000 iterations. These thresholds were chosen by trial and error over many tests where we have observed that the configuration of the MLP (i.e. the number of hidden layers and the flatness of the activation functions) is the primary determinant in the convergence of the training periods. Once properly configured for the task the MLP trains and converges very reliably.

During training, the MLP continues to map inputs to outputs in an uninterrupted fashion, although the mappings produced during a training cycle can be unpredictable. For the initial training period, when the network goes from a randomly initialized state through the first training set, the mappings quickly move from random (but repeatable) to trained and expected. Subsequent retraining periods produce significantly less variation as the network has shorter distances to go. The typical training period comprises several thousand iterations spanning between 0.5 and 2 seconds, and proves barely noticeable to most users. At all other times the network operates as anticipated, efficiently transforming inputs into control outputs (where inputs are received approximately twenty times a second, in our applications).

4. APPLICATIONS

4.1 iPad 1

The simplest interface and implementation consists of an iPad driving a granular synthesis engine. Here we take the two-dimensional touch input (position on the surface) as the feature data, scaling it and feeding it directly to the ART module. This serves to parse the area of the touch screen uniquely for each session and makes a simple and dramatic demonstration of the system. Every time the ART identifies a new region of the feature data (and thus the screen) it is mapped to a new output from the preset granular synthesis outputs. The result is a two dimensional mapping of the nine-dimensional granular synthesis parameters, where any touch on the screen is mapped to a point in the synthesis space.

4.2 iPad 2

We attempt to characterize *drawing style* in the second application, analyzing the iPad touch data for speed of movement, curvature of the line, average direction of movement (taken over sixteen samples) as well as simple position. This results in five parameters that are now mapped through the MLP to the nine-dimensional granular synthesis parameter

space. In this application the user is able to achieve much finer control of the output, mapping many more input areas to outputs. Thus, for example, a straight, slow line in the center of the screen moving to a straight, slow line near the edge produces a gradual sonic change, while a change to a curvy, fast line and back to straight traverses the output parameter space much more dramatically.

4.3 Violin

Providing control to an acoustic musician requires many more dimensions of input. This application utilizes analyzed parameters of the sound (brightness, noisiness, amplitude, and average frequency), as well as musical and textural components (rate of attack and pitch class). Both simple scaling mechanisms and spatial encoding algorithms are employed to produce two separate feature vectors (one for immediate sound parameters and one for a pitch-based short-term memory), which in turn are fed to two distinct ART modules. The outputs of the ARTs serve to train two different MLP networks, one driving the granular synthesis engine and the other controlling real-time graphical animations.

5. CONCLUSIONS

This self-supervising model shows the applicability of unsupervised and supervised ML algorithms working together in an interactive multi-media performance system. All three of our implementations have been employed successfully in demonstrations and performances, and additional testing and development is underway. While the system promises full autonomy for the interactive computer it is currently limited by the pre-definition of the outputs. In this way it functions like supervised ML systems of recent decades. However this may be alleviated by providing the system with methods to analyze and filter the outputs. The parsing of the outputs can be done in a fashion analogous to the inputs, and similarity measures (i.e. various distances between features and categories) can give the computer a path to relating inputs and outputs automatically. Thus a minimal shift in the input music or gestures can be matched with a minimal change in the multi-media output, and significant movements can be treated similarly.

In the current implementations the level of detail set in the ART module has a great effect on the results of the mapping network. When set to produce more general categories (accomplished through a lower vigilance setting) training points are created very far apart in the input feature space. When set to be more precise the training points are put close together. The former gives the performer more gradations of control, but requires large musical shifts to produce noticeable changes in the multi-media system. On the other extreme the smallest changes (changing notes or slight dynamic levels) causes significant changes in the output. Finding a suitable middle ground requires a period of testing to tailor the ART parameters to the music that a given performer desires to play.

While the goal of a fully automatic self supervising system is approaching, the current implementations still demand a noticeable amount of awareness on the part of the user. Although the performer is free to improvise and continually explore new material the mappings will continue to appear new as well. Our experience has shown that this lack of constraints can be challenging. The responsibility is entirely on the performer to remember what they presented to the system and affect its recreation if they desire a return of the same multi-media outputs.

6. REFERENCES

[1] J. Aucouturier and F. Pachet. Tools and architecture for the evaluation of similarity measures: case study of timbre similarity. *Journal of the American Society for Information*, Special Issue on Music Information Recrieval, 2004.

[2] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[3] N. Collins. *Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems*. PhD thesis, University of Cambridge, Cambridge, UK, 2006.

[4] R. B. Dannenberg, B. Thom, and D. Watson. A machine learning approach to musical style recognition. In *Proc. International Computer Music Conference*, pages 344–347, 1997.

[5] C. J. Davis and J. S. Bowers. Contrasting five different theories of letter position coding: Evidence from orthographic similarity effects. *Journal of Experimental Psychology: Human Perception and Performance*, 32(3):535–557, 2006.

[6] P. P. de León and J. Inesta. Pattern recognition approach for music style identification using shallow statistical descriptors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(2):248–257, Feb. 2007.

[7] R. Fiebrink, P. R. Cook, and D. Trueman. Play-along mapping of musical controllers. In *Proceedings of the International Computer Music Conference*, 2009.

[8] R. O. Gjerdingen. Categorization of musical patterns by self-organizing neuronlike networks. *Musical Perception*, 1990.

[9] A. Miller George. The magical number seven, plus or minus two: some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

[10] F. Pachet. Musical data mining for electronic music distribution. *Web Delivering of Music, 2001. Proceedings. First International Conference on*, pages 101–106, Nov. 2001.

[11] I. Peretz and R. J. Zatorre. Brain organization for music processing. *Annual Reviews*, Psychology(56):89–114, 2005.

[12] F. G. P. Piat. *Artist: Adaptive resonance theory to internalize the structure of tonality*. PhD in human development and communication sciences, University of Texas, Dallas, Aug. 1999.

[13] E. Schubert, J. Wolfe, and A. Tarnopolsky. Spectral centroid and timbre in complex, multiple instrumental textures. In *Proceedings of the 8th International Conference on Music Perception and Cognition*, Sydney, Australia, 2004. University of New South Wales.

[14] B. Thom. Interactive improvisational music companionship: a user-modeling approach. *User Modeling and User-Adapted Interaction*, 13:133–177, 2003.

[15] B. Tillmann. Music cognition: Learning, perception, expectations. In *Computer Music Modeling and Retrieval. Sense of Sounds.*, pages 11–33. Springer, Berlin, 2008.

[16] D. Wessel. Connectionist models for musical control of nonlinear dynamical systems. *The Journal of the Acoustical Society of America*, 92(4):2402, Oct. 1992.

Beatscape, a mixed virtual-physical environment for musical ensembles

Aaron Albin
Georgia Tech Center for Music Technology
840 McMillan Street
Atlanta, GA 30332
aalbin3@gatech.edu

Sertan Şentürk
Georgia Tech Center for Music Technology
840 McMillan Street
Atlanta, GA 30332
sertansenturk@gatech.edu

Akito Van Troyer
MIT Media Lab
77 Mass. Ave., E14/E15
Cambridge, MA 02139-4307
nav.reyort@gmail.com

Brian Blosser
Georgia Tech Center for Music Technology
840 McMillan Street
Atlanta, GA 30332
bpb54321@gmail.com

Oliver Jan
Georgia Tech Center for Music Technology
840 McMillan Street
Atlanta, GA 30332
oliverjan@gmail.com

Gil Weinberg
Georgia Tech Center for Music Technology
840 McMillan Street
Atlanta, GA 30332
gilw@gatech.edu

ABSTRACT

A mixed media tool was created that promotes ensemble virtuosity through tight coordination and interdepence in musical performance. Two different types of performers interact with a virtual space using Wii remote and tangible interfaces using the reacTIVision toolkit [11]. One group of performers uses a tangible tabletop interface to place and move sound objects in a virtual environment. The sound objects are represented by visual avatars and have audio samples associated with them. A second set of performers make use of Wii remotes to create triggering waves that can collide with those sound objects. Sound is only produced upon collision of the waves with the sound objects. What results is a performance in which users must negotiate through a physical and virtual space and are positioned to work together to create musical pieces.

Keywords

reacTIVision, processing, ensemble, mixed media, virtualization, tangible, sample

1. INTRODUCTION

Virtuosity, in general, is defined as having advanced skills in a particular or multiple musical areas. We are especially interested in interdependent virtuosity, in which performers demonstrate and improve their skills collectively in some areas such as in a band or a symphony orchestra. We are also interested in exploring novel manners for ensemble interaction that cannot be achieved in traditional acoustic means. Our goal, therefore, is to make novel environments that encourage users to interact with one other and allow for new strategies for interaction. Specifically, we would like to:

1. Create an ensemble that has interdependencies between different performers.
2. Allow for a level of virtuosity, not as a single person but in an ensemble context, like a small chamber orchestra
3. Use physical means of creating the sounds without relying solely on a traditional computer point and click interface while allowing for some degree of gesture control.

2. BACKGROUND

There have been many studies of networked, ensemble, multi-user collaborative frameworks [9, 13]. One of the first examples of this came about through the League of Automatic Composers and later iterations of the Hub, in which users could both produce and alter each others music information [2, 7]. More recent notable examples from which have parallels to our work include the Reactable which is a tabletop interface allowing performers to use a common area, moving physical objects on a surface which are essentially generative audio synthesizers [10]. While the Reactable provides for a very virtuosic ensemble instrument, the interaction with the Reactable is limited to placing objects on a table, which does not allow for richer and more expressive gestural input such as continues hand gestures in 3D.

An example for an interdependent musical instrument is the Tooka, a wind instrument that forces the users to interact with each other by playing on two ends of a hollow tube. Players place opposite ends in their mouths and modulate the pressure in the tube, controlling sound. Coordinated button presses control the music as well, thus tending to create music that explores intimacy and cooperation [5].

The Reactogon is another instrument has the property of interconnections between sound objects causing activities like chain reactions and triggering other objects [3]. It quantizes the space and uses the idea of a harmonic table to allow for easy creation of chords and other musical patterns.

Another environment that uses a physical interface to manipulate virtual sound objects is Drile [1]. Here, musical nodes are represented as worm like objects which can be manipulated and grouped into trees residing in virtual rooms. It is a live looping musical environment by which each worm can be manipulated by scrubbing them through different tunnels causing different sound effects.

Beatscape is similar to these projects with respect to using physical objects to manipulate virtual entities. We do this using gestural and pointing devices as well as tangible interfaces. It also is an environment that encourages groups to work together to achieve musicality. The contribution we

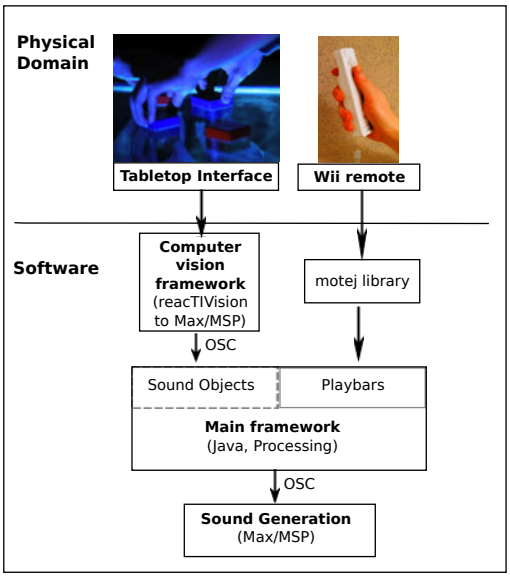


Figure 1: Beatscape Setup

present in this project is the combination of different gestural interactions by two different sets of performers, and the requirement to synchronize gestures based on calculated time delays between gestures and sound triggering.

3. SETUP

In this section we will describe the physical and virtual domains of Beatscape including the hardware and software architectures used to create the environment (Figure 1).

3.1 Physical Domain

Some "conventional" instruments, such as stringed or brass, require the player to skillfully use both hands in order to generate the sound. Each hand has several different techniques to fulfill its role and the synchronized actions of the hands contribute to the pitch content, timbral quality and timing of the sounds. As an example, on a guitar, the left hand typically defines the notes to be played while the right hand causes the sound to come forth. In the guitar case, we usually have one performer controlling both hands; therefore, we need to take account of the interaction between multiple users. In Beatscape we take this paradigm and separate it to two different types of players: those who can place or set up the sound and those who can trigger them.

3.1.1 Tabletop interface

To manipulate the sound objects, we decided to use a tangible table-top interface. We used a set of children's toy blocks as the physical manifestations of the sound objects. Each block has a fiducial marker, specially designed for the reacTIVision framework [11] (explained in Section 3.2.1). The marker is taped to the bottom of the block, placed on a transparent glass table. A camera is placed beneath the table to detect the markers. To quickly identify the sounds during the performance, we drew a picture of the virtual avatar on top of the blocks. The sides of table are covered with dark cloth, so no external light might cause problems. Additionally, two diffuse-light spotlights are placed beneath the table so that they illuminate the markers sufficiently.

3.1.2 Wii remote

To create triggering objects that would be used to collide with the sound objects we decided to use the Nintendo Wii Remote with the motej library [6] to give us some degree of gestural movement. While we considered using a more advanced gestural toolkit or a machine learning mechanism for expressive gestures, we realized that the simple act of pointing using the "sensor bar" and the IR camera attached to the Wii remote was an effective method to identify the specific sound object we wanted to trigger. Triggering was accomplished merely by a simple threshold detection of the acceleration component. More advanced gestures were unnecessary since the Wii remote has buttons and a directional pad allowing us to select different options with ease.

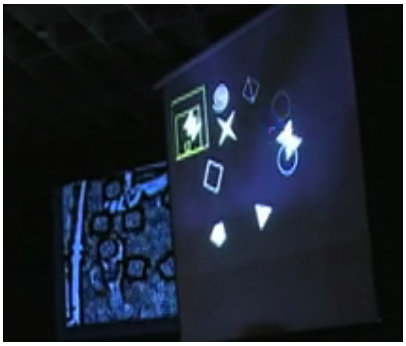


Figure 2: Sound object avatars, one-shot playbars and reacTIVision toolkit projected side by side

3.2 Software Architecture

The software component of Beatscape can be divided into the computer vision framework, the performance framework (Figure 2) built primarily in Java using the Processing API for drawing sound objects and playbars on the screen, and the sound generator realized in Max/MSP.

3.2.1 reacTIVision

We used the reacTIVision toolkit as the computer vision framework, as it enables fast and robust way of tracking fiducial markers. It allowed us to associate the physical blocks tagged with a fiducial marker for each of our sound objects without a need for more advanced image processing. The data is sent through a Max/MSP patch that acts as a mediator between reacTIVision and the main Java application via Open Sound Control (OSC).

3.2.2 Sound Objects

When deciding how the sound objects would be visually displayed for the performers and the audience, we wanted the visual images to be easily associated with the sound and the objects to be animated upon being triggered. We decided to design the visual avatars in Inkscape using SVG format (Figure 3), because this allows sparse representations of simple images and enables us to animate different parts of an image separately. For example, one could animate the eyes nose and mouth of a face image independently. A number of different animations were explored, including rotations, size expansions, and blinking. The avatars are added, moved and removed by the commands received from reacTIVision.

3.2.3 Playbars

We settled upon three basic types of playbar objects that create sound upon collision with sound objects. Each player is represented either by a circle or a square as their base triggering object. They can point on the screen using the IR "sensor bar". In order to deal with the problem of having jittery motions caused by hand movements, we allow the wii mote users to freeze and unfreeze their cursors by pressing

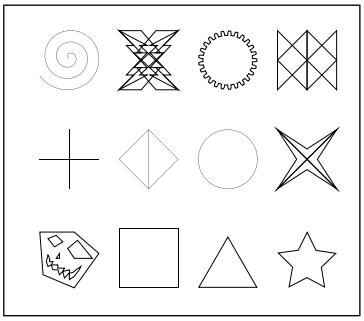


Figure 3: Sound object virtual avatars

the button "A". Then, to create a playbar object, the user must make a jerking motion.

The first type of triggering object is a "one-shot, wavefont-like" playbar where their shape expands from their cursor to some maximum size (Figure 2). The rate of expansion is controlled by the acceleration component. We set a threshold to choose two different rates of expansion: one slow and another fast. The second type of triggering object is similar to the first, except that once it reaches its maximum size, it restarts and repeats its expansion again.

The third type of playbar is a flashlight object in which the cursor is a maximum sized circle or square, filled in. When the flashlight object hovers over a sound object, it causes it to play at its maximum repetition rate. Additionally, by pressing the button "B", the flashlight can be removed from the cursor so that it remains in the spot, allowing the user to create a new flashlight again by making a jerking motion (Figure 4).

A repeating or the flashlight playbar can be selected with the directional pad and deleted by pressing the "-" button. Additionally, for a quick silence mechanism, the "+" button removes all playbars.

3.2.4 Sound Generation

When a sound is triggered, an OSC message containing the sample name and the playback rate is sent to a Max/MSP patch that plays the samples. We use the so-called "gating" mechanism often seen in many hip hop sampler instruments. When a sound is triggered and then re-triggered quickly, the first instance is cut off and the sample is played from the beginning. We ensure that any subsequent trigger of any sample will occur with a inter-onset interval of 100ms at its fastest. This was a decision influenced by our personal aesthetics in hip hop and popular dance and electronic music. Moreover, as an aesthetic decision, the samples are either hard-panned to left and right to achieve spatialization.

4. TECHNIQUES OF THE PERFORMERS

Through our interactions in preparing a piece, a number of different techniques were developed by the Wii remote players as well as the sound object placers.

4.1 Wii remote Players

For the Wii remote players using the one shot bars, they can point to a specific location, lock, and create multiple waves from that point. The players must take into account the time it takes for the waves to expand and collide with a sound object. If they want an immediate triggering, they must point at the object directly and then launch a wave. However, they can also have a "delayed" trigger by creating waves near object, giving a degree of spatio-temporal control over collisions. Another technique is to create a "trailing" wavefront by keeping the cursor unfixed, moving

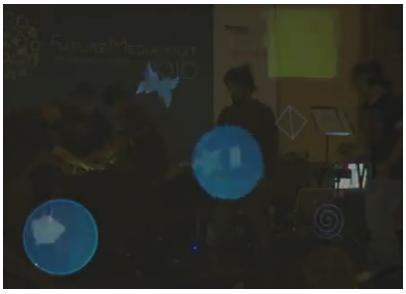


Figure 4: The virtual and physical domains: Virtual interface shows sound and flashlight objects. The sound object players (to the left) and Wii mote players (to the right) are presented in the background

it across the screen and tapping the Wii remote to create new expanding playbars.

With repeating playbars, one technique is for Wiimote players to make "minefields". This allows the sound object players to take control of the expressivity by deciding where to place the sound objects within the expanding fields. Also, by creating multiple repeating bars in the same location at different time intervals, the sound objects will be triggered in complex rhythmical patterns.

With the flashlight object, by hovering over a sound object it will cause the sound to play repeatedly at its maximum rate. By flicking the Wii remote over a sound object, it can strobe the sound object. Another feature of the flashlight is to disassociate it with the cursor and leave it behind for a sound object player to use.

4.2 Sound Object players

Sound object players have a number of techniques. The angle of a sound object is mapped to four discrete playback speeds; thus, these players are responsible for the harmonic and melodic progression of the piece. However, this is dependent upon whether or not there is a playbar to trigger the sound. This interdependency forces different instances for pitch changes: when the playbars are of type single shot, the sound object players have to prepare for rotating the object in advance, whereas in the minefield and the flashlight example, they can change the pitch quicker without need to anticipate as in the previous case.

Sound object players can influence Wiimote players' trajectories by forcing them to decide which route to take if using the "trailing" technique. Putting the objects in proximity to each other allows Wiimote players to simultaneously or consequently trigger multiple sound objects based on relative spacing between objects and playbars. Conversely, in the "minefield" approach, sound object players can decide when to trigger the objects by placing the blocks on and off of the table; the sound players can use this technique to create complex rhythmic patterns.

The sound object players can also give objects character in a limited fashion. For example, the objects can be wiggled both to get the attention of the audience as well as to get the attention of the Wii remote players in order to remind them when to trigger them at certain points in the piece. The objects which have facial or more anthropomorphic features such as the scream/jack-o'-lantern combo seem to have more personality.

5. COMPOSITION AND PERFORMANCES

For our performance, we composed a structured improvisation piece with three distinct sections that were meant

Sample Type	Name of the Sound Objects
Percussive	Snare, Bass Kick, Scratch, Click
Guitar Chords	"High", "Tension", "Bass", "Finish"
Misc. Effects	Scream, "Whoosh", "Synthesizer", "Clap"

Table 1: Types of sound samples used in the performance and sound objects associated with each type

to showcase different techniques. The first section dealt with introducing the environment to the audience and then arranging the sound objects into a grid, putting the emphasis on the Wii remote players. The second section is a "minefield" approach whereby control is handed over to the sound object players. The third section, where both sound objects and playbars are free to move, showcases the flashlight approach. The piece we performed illustrates all the techniques discussed previously.

We contributed sound samples based off of our current listening interests. After collecting various sounds, we organized them into percussive sounds, chord structures, and miscellaneous effects like screams and "whoosh" (Table 1). We used very highly compressed guitar chords from the band Justice, from their debut album *Cross* [4]. All of the percussive sounds and two of the effects are from the Freesound Project [12]. The "clap" and "synthesizer" effects are from FL Studio Legacy Pack [8].

The first performance of Beatscape was in Listening Machines 2010, the annual concert series hosted by Georgia Tech Center for Music Technology, showcasing the work of masters students in the Music Technology program. The performance took place at the Eyedrum in April 2010, in Atlanta, Georgia. The piece was performed again in October, 2010 with Aaron Albin, Sertan Şentürk, Avinash Sastry, Andrew Collela and Sang Won Lee in the FutureMedia Fest 2010, hosted by Georgia Institute of Technology[1]. During the performances we used two screens: one for Reactivision output and the other for our visualization. We decided to show Reactivision for the audience who could not see the table from the back of the hall, so that they would understand how the avatars are controlled.

6. EVALUATION

For the Wii remote players, it was very easy to trigger sounds on the screen and set up different playbars. For the sound object players, the table top interface also proved to be a very intuitive. However the real challenge and skill came about through cooperation both within the Wii remote players and the sound object players so that the performance wouldn't become cluttered. Although we did not conduct user studies to assess the learning curve and the development of skill, by practicing we learned to cooperate with each other and give each other some space, thereby achieving our objective to work along as an ensemble.

The performances were well received. Through informal discussions with audience members, they were seen as self-explanatory and both visually and aurally appealing.

For future iterations, we would like to allow importing any type of image as well a means to associate an arbitrary sound to an image in real time, which could make for an interesting networked variation that adds more spontaneity to the improvisation. We would also like to change the animation with respect to the pitch of the sound object and give the avatars more of a personality for better visualization, feedback to the players and more association to the visual to the audio for the audience. The creation of sound objects can further be associated with custom animations associated with the sounds, giving the sense that they have an intention. Additionally we hop to conduct user studies to further explore collaboration and usability of Beatscape.

[1]Both performances are available online at http://vimeo.com/11676226 and http://vimeo.com/16113180

7. CONCLUSIONS

We have presented Beatscape as a virtual/physical environment that encourages ensemble virtuosity. While sound generation is a simple process of a sound object colliding with a playbar, by separating the tasks among different players, we force the users to work together to create coherent pieces. Thus Beatscape is easy to understand and start out playing but requires a group effort in order to achieve something musically meaningful.

8. ACKNOWLEDGMENTS

We would like to thank Avinash Sastry, Andrew Collella and Sang Won Lee, for assisting with the performance in FutureMedia Fest 2010, Atlanta, GA.

9. REFERENCES

[1] F. Berthaut, M. Desainte-Catherine, and M. Hachet. DRILE: an immersive environment for hierarchical live-looping. In *Proc. of NIME 2010*, pages 192–197, 2010.

[2] J. Bischoff, R. Gold, and J. Horton. Music for an interactive network of microcomputers. *Computer Music Journal*, 2(3):24–29, 1978.

[3] M. Burton. reacTogon. http://www.youtube.com/watch?v=AklKy2NDpqs, 2008.

[4] Cross. Justice. Album, June 2007.

[5] S. S. Fels and F. Vogt. Tooka: Explorations of two person instruments. In *Proc. of NIME '02*, pages 116–121, May 2002.

[6] V. Fritzsch. motej, a slim Java library for Wiimote communication. Sourceforge web-site: http://motej.sourceforge.net/index.html, 2009.

[7] S. Gresham-Lancaster. The aesthetics and history of the Hub: The effects of changing technology on network computer music. *Leonardo Music Journal*, 8:39–44, 1998.

[8] Image-Line. FL Studio Legacy Pack. Software.

[9] S. Jordà. Multi-user instruments: models, examples and promises. In *Proc. of NIME '05*, pages 23–26. National University of Singapore, 2005.

[10] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reacTable*. In *Proc. of ICMC '05*, pages 579–582, 2005.

[11] M. Kaltenbrunner and R. Bencina. reacTIVision: a computer-vision framework for table-based tangible interaction. In *Proc. of the 1st international conference on Tangible and embedded interaction*, pages 69–74. ACM, 2007.

[12] D. Murphy, Raphael "HardPCM" Couturier, M. Carrier, J. Steiner, cdrk, and hgavin. Sound samples. The Freesound Project: www.freesound.org.

[13] G. Weinberg. *Interconnected Musical Networks - Bringing Expression and Thoughtfulness to Collaborative Music Making*. PhD thesis, MIT, 2002.

MoodifierLive: Interactive and collaborative expressive music performance on mobile devices

Marco Fabiani, Gaël Dubus and Roberto Bresin
KTH Royal Institute of Technology
School of Computer Science and Communication
Dept. of Speech, Music and Hearing
Lindstedtsv. 24
100 44 Stockholm, Sweden
{himork,dubus,roberto}@kth.se

ABSTRACT

This paper presents *MoodifierLive*, a mobile phone application for interactive control of rule-based automatic music performance. Five different interaction modes are available, of which one allows for collaborative performances with up to four participants, and two let the user control the expressive performance using expressive hand gestures. Evaluations indicate that the application is interesting, fun to use, and that the gesture modes, especially the one based on data from free expressive gestures, allow for performances whose emotional content matches that of the gesture that produced them.

Keywords

Expressive performance, gesture, collaborative performance, mobile phone

1. INTRODUCTION

In the last five years, the evolution of mobile devices (in particular, mobile phones) and services has been fast and disruptive. Today, a very large part of the population owns a mobile phone, many of which are smartphones, devices that allow, among other things, to connect to the internet, listen to music, and run small applications and games. Following a trend in PC- and console-based video games, several interactive music mobile applications and games appeared that became instant best-sellers (e.g. Smule's *Ocarina*[1] and more recently the *Reactable mobile*[2]).

MoodifierLive is an application that aims at combining on a mobile-platform two different aspects of music-related research: automatic performance, and expressive gesture analysis. Mobile phones were used for their immediate availability, the ability to create an all-in-one solution (i.e. sound production and control device), and their connectivity options. *MoodifierLive* was developed in the contest of the FP7 EU ICT SAME Project[3] (Sound And Music For Everyone Everyday Everywhere Every-way, 2008-2010).

[1] http://www.smule.com/
[2] http://www.reactable.com/products/mobile/
[3] http://www.sameproject.eu/

NIME'11, 30 May–1 June 2011, Oslo, Norway.

The aim of automatic performance is to allow computers to play a musical score (e.g. MIDI file) in an expressive way, by imitating the techniques used by musicians. This is achieved by introducing deviations from the nominal values of acoustical parameters such as a tone's length and dynamic level, and tempo. The KTH rule system for music performance [11] is a large set of rules which define the value of these deviations analytically. The effects of the rules are cumulative, their relative contribution defined by a weighting factor. By changing the value of the weighting factors, one can modify the performance in real-time.

Music performance have a strong connection with movement and gesture [6]. Gestures produced by musicians do not only have a functional purpose (i.e. to produce a specific sound), but also help to convey her expressive intention. Expressive gestures, extracted from video analysis, have been previously used to control the KTH rule system [10]. In *MoodifierLive*, data from the phone's built-in accelerometer are analyzed and mapped to performance macro-rules to obtain corresponding expressive music performances.

2. MOODIFIERLIVE

MoodifierLive is a mobile phone application designed to work on Nokia S60 series mobile phones. It is written in Python, and requires the PyS60 interpreter[4] to be installed on the phone. It plays MIDI files expressively using the KTH rule system for music performance [10, 11] to control the main acoustical parameters of the musical performance (i.e. tempo, dynamics, and articulation).

The use of the performance rules allows even the non-musicians to obtain musically acceptable interpretations of a score, by offering high-level, more intuitive controllable parameters, which are automatically mapped to the low-level acoustical ones. For instance, a classic performance technique, the phrase arch, in which the first part of a musical phrase is played with *crescendo-accelerando* and the second with *decrescendo-ritardando*, is automatically applied by the *phrase-arch* rule (provided that the phrasing has been previously defined in the score file).

The application offers five different interaction modes (described in detail in Sect. 2.1-2.5), two of which give direct control over some of the numerous performance rules, while the remaining three simplify the control even further by introducing an additional mapping, from emotions or expressive gestures to rules.

2.1 Sliders mode

In the sliders mode the user has the ability to control the values of the four main performance rules: *Overall Tempo*

[4] http://www.pys60.org/

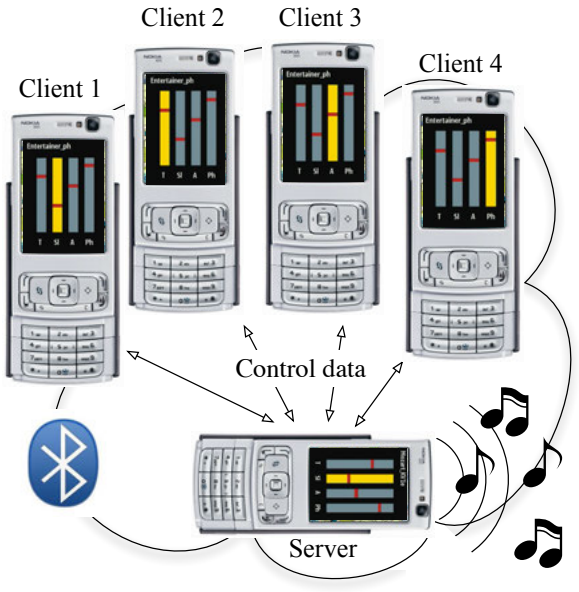


Figure 1: Schematic view of the collaborative interaction mode. Note the displays showing the four sliders controlling Overall Tempo, Overall Sound level, Overall Articulation, and Phrasing.

(T), *Overall Sound level* (Sl), *Overall Articulation* (A), and *Phrasing* (Ph). The sliders mode was designed to allow the naïve user to explore the effect of the single rules on the performance, as well as to allow more advanced users to fine-tune their performances. More rules are available, but to strike a balance between simplicity and usability, we used only the ones that we consider most important to obtain a good performance. Previous research as demonstrated that only tempo, sound level, and articulation account for about 90% of the communication of emotion in expressive music performance [12].

2.2 Collaborative mode

Playing together with other people is an important aspect of the music performance experience, which is both challenging and entertaining. An example of collaborative music performance with mobile phones was proposed in the CaMus^2 project [14]: the camera was used to navigate a marker sheet, and the extracted parameters (e.g. rotation, height) were mapped to MIDI control messages and sent to a PC running MIDI sound synthesis software. Our implementation of a collaborative performance experience allows up to four users to contribute to the creative process by taking control of one or more of the four parameters available in the sliders mode (section 2.1). This simple implementation of a collaborative mode does not aim at reproducing the complex interactions established in an ensemble, but it nevertheless introduces a social component to automatic music performance.

The mode is based on a server-client architecture (see Figure 2.1). In the current implementation, the main application (i.e. *MoodifierLive*) runs on a mobile phone acting as server, which is responsible for the music playback. The client phones (up to four), connected to the server via Bluetooth, are mere control devices. Once a client-server connection is established, the server begins sending regular status-update messages to the clients.

The status-update message is a string of characters, of variable length, beginning with a colon and ending with a semicolon, of the form `:tnn smm aii pjj xxx...x;`. The first twelve characters, divided in four groups of three, correspond to the four parameters (i.e. sliders). The first character in each group (i.e. `t`, `s`, `a`, and `p`) can assume one of three values: `F` if the parameter is free for booking; `B` if the parameter is booked by another client; `S` if the parameter is selected by the client. The second and third character in each group (i.e. `nn`, `mm`, `ii`, `jj`) contain the value of that parameter.The remaining characters (i.e. `xxx...xxx`) contain the name of the current score. For example, the message `:S20B11F59B44The_Entertainer;` means that the tempo parameter is selected by the current client, and has a value of 20; the overall dynamics and phrasing parameters are booked by other clients, and have values 11 and 44, respectively; the articulation parameter is free for booking and has a value of 59; the title of the score is "The Entertainer". The sliders on the client phones are updated with the values in the status messages. Their color indicates if they are booked or free.

The user of a client phone can book and change the value of more than one slider. Any user action generates a command message, which is sent to the server phone. The command message is a string of three characters, opened by a colon, and closed by a semicolon, of the form `:cni;`, where `c` indicates the command: `B` for book; `R` for release; `M` for modify. The second character, `n`, indicates the number of the parameter for which the command was issued (i.e. `0`, `1`, `2`, or `3`). Finally, `i` indicates if the value of the parameter has to be increased (`+`) or decreased (`-`) by a constant step, although only if `c` = `M`. For example, the command `M2+` orders the server to increment the articulation value.

2.3 *Navigate the performance* mode

To simplify the performance control, the use of expressive performances based on macro-rules was introduced [3]. These are sets of low-level rules with a specific set-up that corresponds to a specific emotional expression, for example happiness or sadness. The activity-valence space, previously used in [10], is employed as a simple bi-dimensional model to define different emotions. The activity-valence space boundaries are defined by the sets of values from an experiment [4] in which several expert musicians were asked to create expressive performances of a few musical pieces by setting the values of seven musical variables (tempo, sound level, articulation, phrasing, register, timbre, attack speed). The intermediate values in the bi-dimensional space are obtained by interpolation.

A virtual ball , confined within the screen boundaries, is used to "navigate" in the bi-dimensional space and thus in the space of possible expressive performances. The ball is controlled by tilting the phone, as if it was in a box (the built-in accelerometer is used to determine where the ball is rolling to). The position, size, and color of the ball are used to visually reinforce the perception of the emotion expressed by the music. The size of the ball is directly coupled with the activity, while its color changes following the findings of a previous study [2] that investigated the colors that listeners associate to expressive performances: yellow for happiness, red for anger, blue for sadness, and pink for tenderness.

2.4 *Marbles in a box* mode

Two gesture-based control modes based were designed to allow for a more intuitive interaction with the performance: *Marbles in a box*, and *Free movement* (see section 2.5). Hand gesture information is derived from the acceleration data provided by the phone's built-in accelerometer, and mapped to the activity-valence space (section 2.3).

The *Marbles in a box* is a metaphor to represent the phone as a box containing some marbles, which can be shaken and moved around. The same concept has been used in several other applications (see for example [16, 7]). The energy of the movement, computed as the squared modulus of the acceleration, is directly mapped to the activity, and the tilt of the phone to the valence. Holding the phone high over one's head (positive tilt) should display a positive emotion (feeling "high"), while holding it down parallel to one's leg (negative tilt), should represent a negative emotion (feeling "down"), as previous studies showed [6]. The energy of the movement is also visualized by changing the color of a marble on the phone screen, as in the *Navigate the performance* mode (see section 2.3). Energy and tilt (and thus activity and valence) are sampled at the same frequency as the accelerometer data ($f_s = 30\,\text{Hz}$), and smoothed with a running average over 40 samples (1.3 s).

2.5 *Free movement* mode

The second interaction mode to make use of the accelerometer to detect the user's gestures is called *Free movement*.This mode, unlike the *Marbles in a box*, was developed using data from actual gestures, collected and analyzed in a previous experiment [9]. Eight people were asked to freely perform gestures expressing one of four basic emotions, continuously for 10 seconds. After that, a classification tree was trained using features extracted from these data. The simple regression tree was chosen for its simplicity and because it requires very little computational power, important when it has to be implemented on a mobile device. The choice of a classifier instead of a regressor (i.e. returning continuous values of activity and valence) was dictated by the fact that the training data was categorical.

Two features were chosen to train the classification tree: the velocity and the jerkiness of the gesture, and in particular their Root Mean Square (RMS) values. Although several features from previous studies were considered [1, 11, 13], many were discarded for reasons such as the high correlation between them, and the low sensitivity of the accelerometer ($\pm$2g), which makes the estimation of the relative position of the phone unreliable. The velocity was computed by separately integrating the acceleration in the three directions after subtracting the average over a time window of about 1.3 seconds (to remove the gravity's bias), and then taking the absolute value of the resulting vector. The jerkiness was computed as the derivative of the acceleration [15]. While the velocity gives a rough indication of the energy of the gesture, the jerkiness is an index of how smooth or spiky the movements are.

Before the features extraction, each participant's data was standardized with its mean and standard deviation over all the performances. This followed the observation that the gestures were very similar between participants, but were performed with different intensities.

Cross-validation was used to determine the minimum-cost tree. The resulting tree slightly differs from the one described in [9], because slightly different features and different frame lengths for the averaging were used. The resulting minimum-cost tree is:

```
if Jerkiness (RMS) > 0.86
   ANGRY
else
   if Jerkiness (RMS) > -0.45
      HAPPY
   else
      if Velocity (RMS) > 0.0
         SAD
      else
         TENDER
```

Each one of the four basic emotions is then assigned a fixed value of activity and valence, based on the values obtained in [4], corresponding to the four corners of the activity-valence space in the *Navigate the performance* mode (section 2.3). The features are sampled and smoothed as in the *Marbles in a box* mode. The classification is thus performed 30 times per second.

3. EVALUATION

3.1 Public evaluation

MoodifierLive, developed in the framework of the SAME project [5], was demoed at two public events: the Agora Festival 2009 in Paris (France), and the Festival of Science 2010 in Genova (Italy). Questionnaires were handed out to the visitors in order to collect feedback.

For the version presented in Paris, only three interaction modes (sliders, *Navigate the performance*, and *Marbles in a box*) had been implemented. The response to the questionnaires was positive: the application was judge interesting and fun to use. Critics were directed towards the control: according to the respondents, it could have been made more interesting. This feedback lead to the development of the two other interaction modes, the collaborative and the *Free movement* modes.

The large size of the groups of visitors at the Festival of Science 2010 prevented us from letting people test the application themselves. The groups were only shown a short demonstration. Very few of them completed the questionnaires. For this reason, we decided to evaluate the two modes based on gestures (i.e. *Marbles in a box* and *Free movement*) in a more controlled experiment, carried out at our lab.

3.2 Experimental evaluation

A simple six-tasks experiment, which is described in more detail in [8], was used to evaluate the two gesture-based interaction modes. For each of the two modes, the participants (6M, 7F) were asked to produce, by shaking and moving the phone, three performances that expressed anger, happiness, and sadness. Before each task, they had some time to freely test the application. When ready, they pressed a key to record the performance. After each task, two questions were asked, to be answered on a seven-steps Likert scale (1 = "Not at all", 7 = "Very much"):

1. How successful were you in the task? How much did the performance correspond to the emotion you were supposed to express?
2. How well did your gesture correspond to the emotion you were supposed to express?

At the end of the experiment, the participants were asked to choose their favorite interaction mode. Accelerometer and performance data (i.e. activity and valence) were also logged in a file for later analysis.

Three goals were set for the experiment. First, to verify if the participants, without any explanation about the mapping from gestures to performance, would understand how to obtain the requested expressive performances. Second, to find out which mode worked better, and which was preferred by the participants. Third, to verify that the gestures corresponded to the emotion expressed by the performance.

The analysis of the log data showed that in the case of the *Marbles in a box* mode, the participants did not understand the connection between tilt and valence (see section 2.5): most of the time, they held the phone horizontal,

Table 1: Mean and standard deviation (Std) of the answers to questions 1 and 2 for the different modes. The questions were answered on a seven-steps Likert scale (1 = "Not at all", 7 = "Very much")

	Marbles		*Free*		Overall	
	Mean	Std	Mean	Std	Mean	Std
Q1	4.56	1.60	5.10	1.47	4.83	1.55
Q2	4.82	1.48	5.51	1.30	5.17	1.43

which resulted in a valence value around zero. On the other hand, the log data for the *Free movement* mode showed that the participants obtained a much wider range of values for activity and valence. This can be in part explained by the fact that, in this mode, the output from the classifier is discrete. Nevertheless, much better separation between emotions was obtained in the *Free movement* mode. This might reflect the fact that this mode was based on the analysis of free expressive gestures, and thus was more natural to understand.

The results from the log data were also reflected in the answers to the questions, which showed a strong preference (92%) for the *Free movement* mode. Statistical analysis of the answers to the second question also revealed that, according to the participants' perception, the agreement between the gestures and the emotions was significantly higher for the *Free movement* mode ($F(1, 12) = 8.748$, $p = 0.012$). All in all, the answers to the two questions revealed that the participants judged both modes to work relatively well (see Table 1).

4. CONCLUSIONS

We presented here *MoodifierLive*, an application in which findings from several previous studies have been implemented on a handheld device. The application allows for interactive control of rule-based automatic music performance, through five interaction modes, of which two based on gestures, and one collaborative. Results from evaluation showed that the application is interesting, fun and relatively intuitive to use.

The limits of the mobile phones used to test the application (i.e. Nokia N95), and specifically of the built-in accelerometer, were the reasons behind some design choices, especially the use of a simple classification tree for gesture recognition, based on a very limited number of features.

The use of more powerful devices would allow us to implement more advanced solutions to several problems. In the *Free movement* mode, a better classifier, which takes into consideration also the time evolution of gestures (e.g. [1]), would improve the expressive possibilities of the system. Furthermore, an automatic calibration of the gesture range would also improve the response of the system to a specific user.

A demo video showing the functionalities of *MoodifierLive* can be found at: `http://www.youtube.com/watch?v=m_9TMmTpjAw`.

5. ACKNOWLEDGMENTS

This study was partially funded by the Swedish Research Council (Grant Nr. 2010-4654), and by the EU SAME project (FP7-ICT-STREP-215749):
`http://www.sameproject.eu/`

6. REFERENCES

[1] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. Continuous realtime gesture following and recognition. In S. Kopp and I. Wachsmuth, editors, *Gesture in Embodied Communication and Human-Computer Interaction*, volume 5934 of *LNAI*, pages 73–84. Springer, Heidelberg, 2010.

[2] R. Bresin. What is the color of that music performance? In *Proc. Int. Computer Music Conf. (ICMC2005)*, pages 367–370, Barcelona, Spain, 2005.

[3] R. Bresin and A. Friberg. Emotional coloring of computer-controlled music performances. *Computer Music J.*, 24(4):44–63, 2000.

[4] R. Bresin and A. Friberg. Emotion rendering in music: Range and characteristic values of seven musical variables. *CORTEX*, 2011, accepted for publication.

[5] A. Camurri, G. Volpe, H. Vinet, R. Bresin, E. Maestre, L. Javier, J. Kleimola, V. Välimäki, and J. Seppanen. User-centric context-aware mobile applications for embodied music listening. In *Proc. of the 1st International ICST conference on User Centric Media*, 2009.

[6] S. Dahl and A. Friberg. Visual perception of expressiveness in musicians' body movements. *Music Perception*, 24(5):433–454, 2007.

[7] A. DeWitt and R. Bresin. Sound design for affective interaction. In A. C. Paiva, R. Prada, and R. W. Picard, editors, *Proc. Affective computing and intelligent interaction (ACII2007)*, volume 4738 of *LNCS*, pages 523–533. Springer, Berlin / Heidelberg, 2007.

[8] M. Fabiani, R. Bresin, and G. Dubus. Sonification of emotional expressive gestures with automatic music performance on mobile devices. *J. Multimodal User Interfaces*, 2011, Submitted.

[9] M. Fabiani, G. Dubus, and R. Bresin. Interactive sonification of emotionally expressive gestures by means of music performance. In R. Bresin, T. Hermann, and A. Hunt, editors, *Proc. ISon 2010 - Interactive Sonification Workshop*, 2010.

[10] A. Friberg. pDM: an expressive sequencer with real-time control of the KTH music-performance rules. *Computer Music J.*, 30(1):37–48, 2006.

[11] A. Friberg, R. Bresin, and J. Sundberg. Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology, Special Issue on Music Performance*, 2(2-3):145–161, 2006.

[12] P. N. Juslin and P. Laukka. Communication of emotions in vocal expression and music performance: Different channels, same code? *Psychological Bulletin*, 129(5):770–814, 2003.

[13] M. Mancini, G. Varni, J. Kleimola, G. Volpe, and A. Camurri. Human movement expressivity for mobile active music listening. *Journal on Multimodal User Interfaces*, 4:27–35, 2010.

[14] M. Rohs and G. Essl. Camus^2 - collaborative music performance with mobile camera phones. In *Proc. Int. Conf. Advances in Computer Entertainment Technology (ACE2007)*, Salzburg, Austria, 2007.

[15] K. Schneider and R. F. Zernicke. Jerk-cost modulation during the practice of rapid arm movements. *Biological Cybernetics*, 60(3):221–230, January 1989.

[16] J. Williamson, R. Murray-Smith, and S. Hughes. Shoogle: Multimodal excitatory interfaces on mobile devices. In *Proc. Computer Human Interaction Conf. (CHI2007)*, San Jose, CA, USA, 2007.

A Physically Based Sound Space for Procedural Agents

Benjamin Schroeder
The Ohio State University
Dept. of Computer Science
and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210
benschroeder@acm.org

Marc Ainger
The Ohio State University
School of Music
110 Weigel Hall
1866 College Road
Columbus, OH 43210
ainger.1@osu.edu

Richard Parent
The Ohio State University
Dept. of Computer Science
and Engineering
395 Dreese Laboratories
2015 Neil Avenue
Columbus, OH 43210
parent@cse.ohio-state.edu

ABSTRACT

Physically based sound models have a "natural" setting in dimensional space: a physical model has a shape and an extent and can be given a position relative to other models. In our experimental system, we place procedurally animated agents in a world of spatially situated physical models. The agents move in the same space as the models and can interact with them, playing the models and changing their configuration. The result is an ever-varying audiovisual landscape.

This can be seen as purely generative—as a method for creating algorithmic music—or as a way to create instruments that change autonomously as a human plays them. A third perspective is in between these two: agents and humans can cooperate or compete to produce a gamelike or interactive experience.

Keywords

Physically based sound, behavioral animation, agents

1. INTRODUCTION

Physically based sound models [4] can be used to create synthetic instruments that react in expressive ways to varied input. Such models often are defined in physical, spatial terms; for example, a plucked string might have a length defined in meters or an acoustic tube a certain diameter. It is therefore natural to think of creating an instrument by arranging these models in some kind of spatial setting. A performer might then use gestural input to play the virtual instrument [7].

Physical models produce realistic sounds, but because they are computer models, they are not limited to the strictly physical. One way to extend the capabilities of a virtual instrument is by introducing procedurally animated *agents* into the same spatial environment as the models. These agents can affect the physical constructs represented by the models in ways impossible or difficult in the real world. For example, the agents could play an instrument in algorithmic ways, but could also change the instrument over time, producing a changing experience for a human performer. Because the models and the agents are spatially situated, the changing system can be appreciated not only in terms of its sound, but also in terms of its visual appearance.

Figure 1: Agents smoothly changing the length of strings being played by a human performer, producing a *glissando* effect.

Figure 1 shows a scenario in which a human performer is playing several strings by strumming them using a mouse gesture. Meanwhile, several agents have grabbed the ends of the strings and are moving, choosing new lengths for the strings at random. This produces an unpredictable *glissando* effect as the strings' lengths, and thus their pitches, change.

In the remainder of this paper, we discuss an experimental system for situating behavioral agents and physical models in the same spatial world. After some background, we present the physical models and their embedding into space, and discuss how agents can interact with and change the models. We conclude with several example scenarios. The discussion throughout is in the context of our proof of concept implementation, which runs in real time on standard Macintosh hardware[1].

2. BACKGROUND

Other musical systems, such as the various modes of Electroplankton[2], have used behavioral motion to produce sound. A recent example is Lush [3], which assigned musical notes to individual flocking agents, giving users a movable play-

NIME'11, 30 May–1 June 2011, Oslo, Norway.

[1] A limited number of models can be simulated in real time; we are confident that increasing processing speeds and advanced simulation techniques will lead to greater capacity in the future.

[2] *Electroplankton* is a Nintendo DS game designed by Toshio Iwai; it was originally released in April 2005.

head which could capture and play the notes. In our system, the environment rather than the agents generates sound, although different agents could (for example) represent different excitation models.

There is a rich literature in computer animation regarding behavioral agents. A good starting place is the work of Craig Reynolds, inventor of the classic "boids" flocking algorithm [6]. We do not impose any particular high-level behavior on the agents in our system, but instead provide an approach for relating agents to physical models.

Cook's book [4] gives a good introduction to physical modeling synthesis. Our proof of concept implementation used finite difference time domain models; many issues surrounding this type of model are discussed in a recent book by Bilbao [1].

The recent web artwork *Conductor*[3], by Alexander Chen, represents subway traffic using physically-inspired strings drawn by moving agents. The sound played by the strings, however, is based on recorded samples.

Schroeder et al. [7] also embedded physical models in space; this work is an evolution of the system they describe. Both systems owe much to the way in which the Reactable [5] used spatial arrangement of its models, although the models in that system, and the particulars of their arrangement, are substantially different from those in the present work.

3. THE SOUND SPACE

The agents in our system interact with spatially situated physical models. Our proof of concept implementation includes models of musical strings and rectangular plates situated in 2D space. The models are based on finite differences [1] and are simulated in real time. These choices were made for the sake of simplicity, but other models could be used as long as they could be situated in space and respond to force input; similarly, the models could be embedded in 3D space.

Each model is described below in terms of its vibratory behavior, which is simulated to produce sound, and also in terms of its embedding in the agent space. The capabilities of agents to change or excite the models are described in a later section.

3.1 String Model

Our string model is an ideal string with basic damping and force-based input:

$$y_{tt} = c^2 y_{xx} - \sigma y_t + f(x,t),\ 0 \le x \le L.$$

In this notation, a subscript indicates a partial derivative; thus y_{tt} is the second time derivative or acceleration of the string.

In the equation above, the coefficient c is the speed of sound on the string; σ is a damping coefficient; force input is given by the function f, which varies in space and time. The string is of length L. Strings are assumed to have fixed boundary conditions.

A string is embedded in the space with a center position, a length, and a rotation. This implies the existence of two endpoints which agents may move; the string's length L is recalculated throughout the simulation based on its endpoints' positions.

3.2 Plate Model

Our plate model is similar to the string model, but differs in two ways. First, the motion of a plate in our system is mainly due to stiffness rather than to tension, and therefore the initial term of the equation of motion is a fourth derivative. Second, the model includes a frequency-dependent damping term meant to mimic the effects of different materials. A similar term could be included in the string model, but it has a greater effect here, allowing for mimicry of materials as distinct as metal and wood.

For the plate, then,

$$u_{tt} = -\kappa^2 \nabla^4 u - \sigma u_t + b_3 (\nabla^2 u)_t + f(x,y,t),$$
$$0 \le x \le L_x,\ 0 \le y \le L_y.$$

Here, ∇^2 is the 2D Laplacian operator, $\nabla^2(u) = u_{xx} + u_{yy}$, and ∇^4 is the biharmonic, $\nabla^4(u) = \nabla^2(\nabla^2(u))$. The coefficient κ describes the plate's stiffness and σ basic damping across all frequencies. The term with coefficient b_3 describes frequency-dependent damping. As with strings, plates are assumed to have fixed boundary conditions at all edges.

A plate is embedded in 2D space in a way similar to the string, above, except that the plate's extent is in two dimensions. Agents may change the extent of a plate.

Plates in our proof of concept implementation are always rectangular. There is no reason in general why non-rectangular shapes could not be used, perhaps through the judicious use of boundary conditions.

4. PROCEDURAL AGENTS

Procedural agents exist in the same space as a collection of sounding models. Each agent is essentially an oriented particle; it has a *position*, a *heading*, and a *velocity*. In order to allow environmental forces to act on the agents, each agent is also assigned a *mass*.

Agents' motion is simulated forward through time based on these properties. Navigation is specified at a higher level and is discussed briefly below.

An agent can affect its local environment in several different ways. First, an agent may apply *excitational force*, such as plucking or tapping, to a nearby sound model. An important variant of this is to feed some external sound signal, such as that of a microphone, into a model by adapting the signal as force input. Similarly, an agent may apply *damping* to a nearby model, allowing it to do such things as fret a string. The particular kinds of forces applied are up to the agent and may depend on factors such as its velocity as well as higher-level behavioral concepts.

An agent may also change the configuration of a nearby model. It may grab a model for *translation* in space or for *rotation* around a fixed point (such as the model's center). A grabbed model travels with the agent until it is dropped.

Finally, an agent may *create* or *destroy* models or other agents. (In our proof of concept implementation, we do not implement coupling between models, but in general an agent could also create or destroy such coupling connections.)

4.1 Model Visibility

An agent has limited access to its local environment. It can "see" and affect models at its current position or that it has crossed in the last time step. In planning navigation, high-level behaviors might make use of additional factors such as which other agents are nearby or which models are some distance ahead of a given agent.

One of the advantages of physical models is that they respond differently to forces applied at different points. A plate resonates differently, for example, with input near its edges than it does with input near its center. Therefore when an agent is given access to a model, the access is through a *proxy* that represents the particular region near the agent. Excitation and damping are then applied through the proxy to affect the model at the appropriate

[3] At http://www.mta.me; retrieved on February 2, 2011.

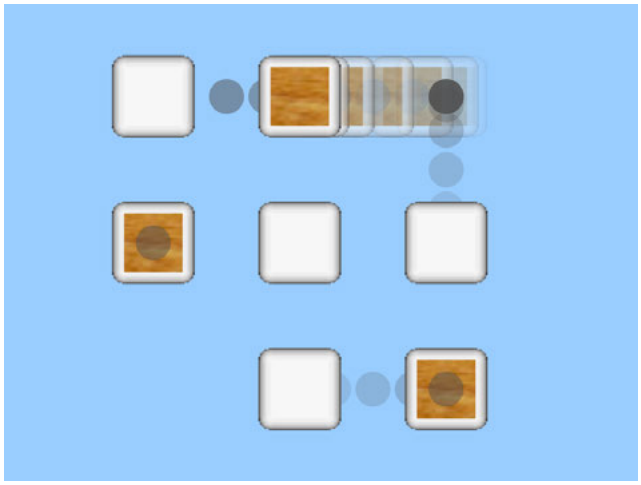


Figure 2: Agents moving and tapping rhythmically on sliding tiles.

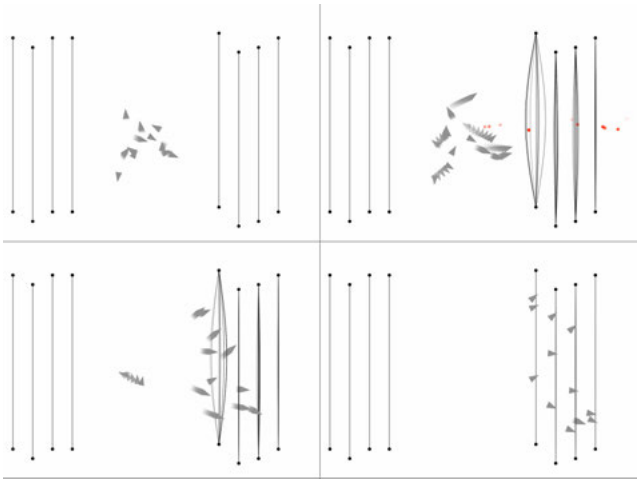


Figure 3: A system in which agents move to damp sounding strings.

location.

Shape changes are implemented as a variant of this: an agent is presented with proxies representing a string's endpoints or a plate's edges, which may then be moved as if they were entire models. Moving the proxies causes changes in the underlying models.

4.2 High-Level Navigation

Various high-level navigational strategies may be used with the agents described above; some of these are described in a later discussion of example scenarios. In general, a navigational strategy is responsible for planning the large-scale motion and behavior of an agent. A trivial example of a behavior might be that of a "water bug":

```
Given an environment with several strings,
Swing around to a random heading;
Skim across the surface at that heading.
If you cross a string while moving,
   Pluck it.
```

In this case, the navigational strategy would implement the logic above and apply forces to the bug to produce the "skimming" movement.

In general, navigational strategies might take the environment into account, sending agents towards a model, for example, or directing an agent to wander up and down the length of a string.

4.3 Environmental Forces

Environmental forces may be used to affect the agents by giving them physics-based motion. Examples of such forces might include gravity wells, repulsive barriers, and directional "wind". These forces change agents' velocities and may be applied based on proximity or globally.

If forces such as these are used in conjunction with other high-level navigation, the navigational strategy is responsible for weighting its input and that of the environment to produce whatever effect is desired.

4.4 Human Agency

A human user may interact with the environment at the same time as the agents, doing anything that an agent can do: playing sound models, moving them around or changing them, or creating or destroying models.

The human's relationship to the agents depends on the particular scenario implemented. For example, agents might change parts of an instrument that a human is playing through gestural input or through the microphone, giving the effect of playing an instrument which is changing over time. On the other hand, a human might interact with the agents more directly, putting obstacles in their way, giving them sound models to play, or guiding them toward a goal, producing a gamelike scenario.

5. EXAMPLE SCENARIOS

We have already seen a simple example of agents producing a *glissando* effect (Figure 1). Below are several other small scenarios involving agents, various behaviors, and physical models.

5.1 Rhythm on Sliding Tiles

Figure 2 shows a few agents moving rhythmically on a playing board made up of several sounding plates. The simulation proceeds as a series of *moves* made by the agents. For any move an agent may choose to

1. Stay still, doing nothing
2. Hop on its present tile, producing a sound
3. Hop to an adjacent tile
4. Slide its tile into a nearby gap
5. Change the material of its tile to be metal or wood

Agents are more likely than not to stay still (otherwise the rhythm would descend into chaos). An agent may only hop onto a tile which is not already occupied.

5.2 Sound and Silence

Figure 3 depicts two sets of strings, each tuned to produce notes from a pleasant chord. When the system is at rest, several agents wait in the middle, making only small, random movements.

The user may play any of the strings. When a string is sounding, agents will depart from the middle and latch onto the string in order to dampen the sound. Some time after a string becomes quiet, an agent damping that string will release the string and return to the middle of the screen.

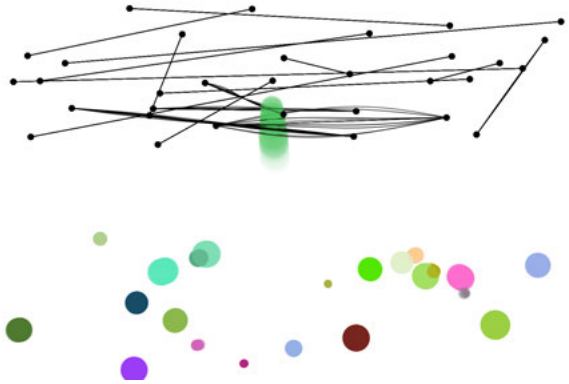


Figure 4: Bubbles representing recordings drift up, releasing their sound into resonating objects.

5.3 Resonance and Motion

Figure 4 depicts an interactive system. Users of the system speak into the microphone in order to record sound bubbles which then cluster near the bottom of the screen. Each bubble is a separate agent; when clustering, the bubbles drift randomly. Longer recordings are assigned larger bubbles.

At random times, a bubble is chosen to be released. The bubble/agent then drifts upwards; it feeds its stored sound into any nearby sound models, causing them to resonate, as it goes. Users interact with this system through the microphone, but a possible enhancement would be to give the users portable gravity and anti-gravity wells to affect the motion of the agents.

6. CONCLUSIONS

We have described an experimental system in which agents and physically based sound models are placed in the same space and allowed to interact. This extends the capabilities of the sound models, allowing for the creation of algorithmic audio or playful interactive systems.

Several questions and opportunities for future work remain. The agents have only a limited kind of interaction with the models; they can feed forces into the models, but are not themselves affected by model motion. It would be interesting to allow the agents to be affected by the sound vibration as well; in that case agents could represent, for example, elements reminiscent of those in the prepared piano [2], or could bounce off of vibrating membranes as if they were trampolines.

Our system does not yet allow for high-level, interactive programming of the behaviors. Such a programming system would be a useful addition, allowing users to experiment with different behaviors and configurations at runtime.

The agent behaviors discussed here are only a start. Further experimentation and research is needed to determine additional interesting musical uses for a behavioral system such as the one described here.

7. REFERENCES

[1] S. Bilbao. *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. Wiley Publishing, 2009.

[2] S. Bilbao and J. ffitch. Prepared Piano Sound Synthesis. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, pages 77–82, Montreal, Quebec, Canada, Sept. 18–20 2006.

[3] H. Choi and G. Wang. LUSH: An Organic Eco+Music System. In *Proceedings of NIME 2010*, 2010.

[4] P. R. Cook. *Real Sound Synthesis for Interactive Applications*. A. K. Peters, Ltd., Natick, MA, USA, 2002.

[5] M. Kaltenbrunner, S. Jorda, G. Geiger, and M. Alonso. The reacTable*: A collaborative musical instrument. *Enabling Technologies, IEEE International Workshops on*, pages 406–411, 2006.

[6] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA, 1987. ACM.

[7] B. Schroeder, M. Ainger, and R. Parent. An Audiovisual Workspace for Physical Models. In *Proceedings of Sound and Music Computing 2010*, 2010.

Acquisition and study of blowing pressure profiles in recorder playing

Francisco García[†]
pul.editions@gmail.com

Leny Vinceslas[†]
leny.vinceslas@gmail.com

Josep Tubau
pi2dlor@gmail.com

Esteban Maestre[†‡]
esteban.maestre@upf.edu
esteban@ccrma.stanford.edu

[†] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[‡] Center for Computer Research in Music and Acoustics, Stanford University, USA

ABSTRACT

This paper presents a study of blowing pressure profiles acquired from recorder playing. Blowing pressure signals are captured from real performance by means of a a low-intrusiveness acquisition system constructed around commercial pressure sensors based on piezoelectric transducers. An alto recorder was mechanically modified by a luthier to allow the measurement and connection of sensors while respecting playability and intrusiveness. A multi-modal database including aligned blowing pressure and sound signals is constructed from real practice, covering the performance space by considering different fundamental frequencies, dynamics, articulations and note durations. Once signals were pre-processed and segmented, a set of temporal envelope features were defined as a basis for studying and constructing a simplified model of blowing pressure profiles in different performance contexts.

Keywords

Instrumental gesture, recorder, wind instrument, blowing pressure, multi-modal data.

1. INTRODUCTION

The process of music performance offers great opportunities for pursuing research on instrumental gestures when investigated from a computational approach based on data observation and analysis. Within the process, the musical message is represented as a written score containing an ordered sequence of note events and annotations of discrete nature. The performer interprets the score and transforms it into a set of physical actions of continuous nature intended to serve as controls for the musical instrument. Those are called instrumental gestures [4]. Furthermore, in the case of excitation-continuous musical instruments (e.g., bowed strings or wind instruments) the complexity of interaction makes the problem becoming much more interesting [4, 7, 6].

In recorder playing, the blowing pressure is often seen as the most important instrumental gesture parameter modulated during performance. The recorder could be considered among the simplest excitation-continuous musical instruments, but the study of instrumental gesture parameters from real performance have been strongly limited by the intrusiveness resulting from a range of measurement techniques, all based on the introduction of plastic tubes (or *catheters*) in the mouth of a performer while playing. Moreover, the direct measurement of blowing pressure signals in flute-like instruments have been limited to the transverse flute with many different studies carried out in the recent history for the extraction of respiratory parameters during performance [1], in the study of performance techniques [6, 2] or in the analysis of frequency content of the breath pressure [8]. The main drawback of these approaches is the intrusiveness of the measurement: the performer is forced to modify his natural performance in order to adapt to the modified instrument. It is difficult to find studies dealing with deep instrument modifications resulting in a reduced intrusiveness or enhanced measurement accuracy, mainly because altering the instrument structure could easily lead to a modification of the timbre.

The remainder of the paper is organized as follows. Section 2 provides an overview of the previous related work. In Section 3, we introduce the acquisition device and setup, the construction of the database and data pre-processing. Section 4 presents the definition of the envelope features and a series of analyses regarding the relation of blowing pressure profile features and performance contexts. Finally, section 5 concludes by summarizing important results and shedding some light on the imminent future work.

2. BLOWING PRESSURE

The recorder belongs to the family of the aerophones, which produce sound primarily by causing a body of air to vibrate without the use of membranes nor strings. Normally, recorders are made up of three separable sections: the head, the middle and the foot piece. The head is the responsible for the primary sound production.

Two main acoustic models try to explain the complex phenomena happening in flute-like instruments: the jet-drive model by Fletcher [3] and the discrete vortex model by Verge [10]. The former neglects many details of the flow at the lip labium which appear to be fundamental for the performance of the instrument [9]. The latter basically describes the timbre of the instrument as a function of the dimensionless velocity of the air jet and the mouth geometry and throws a very important conclusion about the energy transformations that happen during the sound production process: from the pneumatic energy, coming from the air jet developed by the player, 95% is dissipated in the *mixing*

NIME'11, 30 May–1 June 2011, Oslo, Norway.

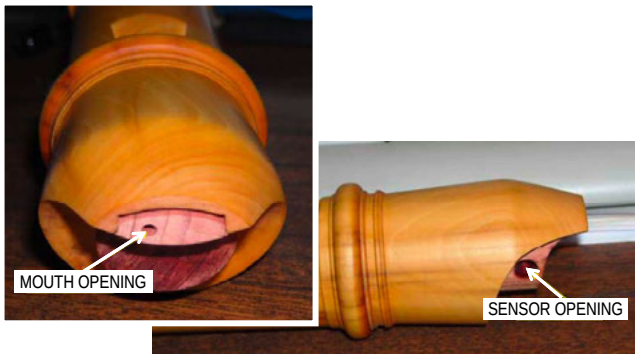


Figure 1: Detail of the openings in the mouthpiece.

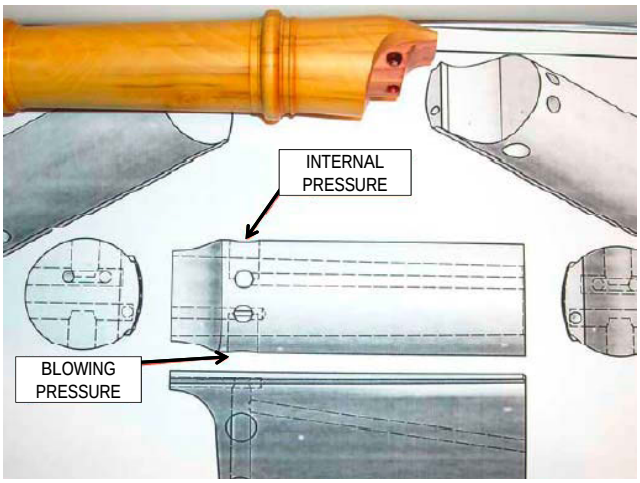


Figure 2: Further detail of the modified mouthpiece, where both ducts can be observed.

region (a concept introduced by Elder in 1973 which refers to the coupling zone at the exit of the windway and the lip labium), i.e. the shedding of vortices at the edge of the labium causes 95% of the energy to dissipate. From the remaining 5% that is transferred to the acoustic oscillation of the air in the pipe, around 3 or 4% is dissipated in viscous and thermal losses to the pipe walls, so that only about 1% of the initial pneumatic energy is radiated as sound. Therefore, we consider that aiming at extracting blowing pressure by measurements carried out after the mixing region would lead to less representative correlates of instrumental control.

With regard to the main instrumental gesture parameters modulating perceptual attributes of the produced sound, the blowing pressure and the fingering could be considered as the most important. Indeed, blowing pressure, as opposed to fingering, presents a continuous nature and allows the control of dynamics, timbre and overblowing techniques. During performance, blowing pressure is exponentially related to pitch [5], and linearly related to the dynamics, although it has not been quantified in an empirical way [6]. Variations in dynamics are achieved through a change in the blowing pressure, although this phenomenon also causes the pitch to slightly change. Fingering allows the performer to alter fundamental frequency and, in combination with blowing pressure, may also help to modulate dynamics. Montgermont [6] studied the relationship between dynamics and blowing pressure for the transverse flute. For a given pitch, the amount of blowing pressure needed for achieving a higher dynamic is obviously higher. Furthermore, the relationship between blowing pressure and fundamental frequency in the case of the transverse flute follows a linear relationship $P = 0.8 \times f$ [3, 6]. In this work, we focus on traditional performance techniques (articulation and dynamics), assuming a fixed fingering position for each of the analysis contexts.

3. DATA ACQUISITION

Blowing pressure and radiated sound are synchronously acquired from real practice by means of a novel, low-intrusiveness measurement setup based on a modified recorder and a close-field microphone. A set of scripts was designed in order to cover a number of performance contexts when constructing a multi-modal database.

3.1 Acquisition of blowing pressure

Special mechanical alterations were carried out in the mouthpiece block of an alto recorder that allowed the connection of two pressure sensors without altering the timbre of the original instrument: the intrusiveness was significantly reduced as compared to using a plastic catheter in the mouth of the musician, i.e. the performer could play the instrument naturally. The instrument was designed by the catalan luthier Josep Tubau, who carried out the modifications and tests under our supervision with the aim of establishing two measurement points: (i) pressure in the mouth of the performer (or *blowing* pressure), and (ii) pressure at the closer end of the resonator pipe (or *internal* pressure).

The first measurement is achieved by means of a connecting duct that joins a hole at the mouth opening and another hole at one side of the mouthpiece, as it is shown in Figures 1 and 2. The second measurement is carried out thanks to an analogous technique, this time relying on a connecting duct with one of its openings at the closer end of the resonator pipe, as it is also shown by Figure 2. The second measurement, while very useful for studying sound production mechanisms, is not used in this work.

As for the pressure sensors, which had to provide a dynamic range of approximately $3000Pa$, we selected a piezoresistive transducer because of a great accuracy together with a small size. The chosen model was the Honeywell© ACSX 01DN. The signal coming from the sensor was acquired using a National Instruments© acquisition card (USB-6009), which provides a sampling rate of $48000Hz$ to be multiplexed among the number of signals to be acquired. In our case, two signals were acquired: the blowing pressure signal, and an audio metronome used later for synchronization with the sound signal acquired with the microphone (see below).

3.2 Data pre-processing

Because of different sampling rates and time-propagated sample period inaccuracies due both to hardware and software typical issues, the audio signal coming from the microphone and the pressure signal coming from the acquisition card had to be re-synchronized. For that purpose, a external audio metronome click signal was recorded both by the audio acquisition device and by one of the channels of the USB-6009 analog acquisition card. By means of a pulse detection algorithm devised for this purpose, metronome clicks were correctly detected from both metronome signals, and the obtained time stamps were used for resampling and synchronizing both signals [8].

The second step consisted on removing the high-frequency component of the acquired pressure signal (at a frequency equal to that of the note being played). The blowing pressure presents a coupling frequency component strongly de-

pending on the effective length of the resonator pipe, i.e. it depends on the fingering of the performer (as it happens with the pressure at the mouth). Filtering of this high-frequency component was achieved by means of numerical smoothing, using a quadratic-regression filter conveniently applied to the signal in order to avoid blurring pressure onsets and offsets as it would happen if a low-pass filtering were used.

The final step consisted in segmenting blowing pressure signals into single notes. For that purpose, a two-stage automatic segmentation technique was developed. First, onset candidates are generated for each note, mostly based on the absolute values of blowing pressure and its first three derivatives. Then an adaptive algorithm, making use of the nominal score and taking into account a maximum deviation of the performer, evaluates which of the generated onset candidates best matches a real onset. Resulting segmentations were manually revised in order to avoid errors in further analyses.

3.3 Database structure

A multi-modal database, including aligned and segmented pressure signals and produced sound, was constructed after carrying data acquisition and processing from a number of recordings with a professional recorder player. A set of recording scripts (mainly musical exercises in the shape of repetitions and scales) was designed so that a balanced set of performance techniques is covered. Four main dimensions (or *performance context* parameters) were taken into account, leading to a total of around 10000 notes. The first analyzed dimension is the **pitch**. The recordings covered the whole tessitura of the instrument, in jumps of 2 or 3 semitones. Each pitch was performed by using a unique (the most common, according to the performer) fingering position. Second, the **dynamics** were divided into three levels: pianissimo (pp), mezzo-forte (mf), and fortissimo (ff). Third, five different **note durations** were recorded. These durations correspond to the duration of a quarter, eighth and sixteenth note at 90 BPM, and an eighth and sixteenth note at 120 BPM, respectively. Finally, four **articulations** (primarily regarded as 'tonguings' by the musician) were considered and labeled as *full legato* (no tonguing, but just diaphragm-driven blowing pressure oscillations), *legato*, *soft staccato*, and *staccato*.

4. DATA ANALYSIS

Data analysis first consisted in the observation of segmented blowing pressure profiles and the identification of a number of envelope features as a basis of further systematic analysis and modeling in different performance contexts.

4.1 Envelope model

In order to devise an envelope model able to consistently represent profiles in different performance contexts, the first step was to observe the blowing pressure envelopes. Figure 3 shows a general picture of the acquired envelopes: three different dynamics for each given articulation and pitch, all of them for the same note duration (also, only three different pitch values -one per octave- are shown). As a first clear observation, each articulation presents a characteristic shape, as a result of different tonguing (when existing). Secondly, the maximum value of blowing pressure reached within each note is positive-correlated with fundamental frequency and dynamics, as it happened for the transverse flute [6]. For the case of legato articulations (uninterrupted air jet, no tonguing) one can observe how the blowing pressure never falls down to the bottom line of 0 Pa, as opposed to what happens with the staccato-like articulations, for which the

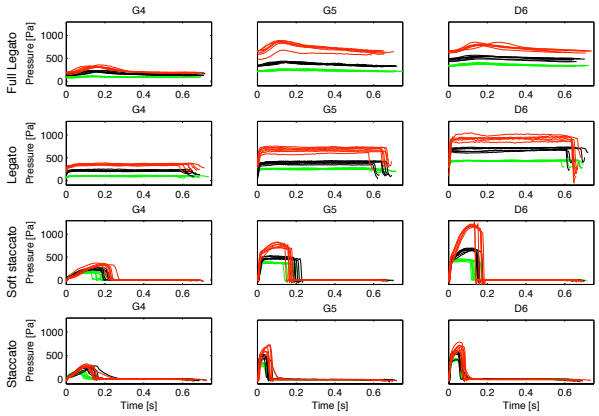


Figure 3: Blowing pressure profiles for different articulations (rows), fundamental frequency (columns) and dynamics (GREEN: pp, BLACK: mf, RED: ff); nominal note duration was 0.66 seconds.

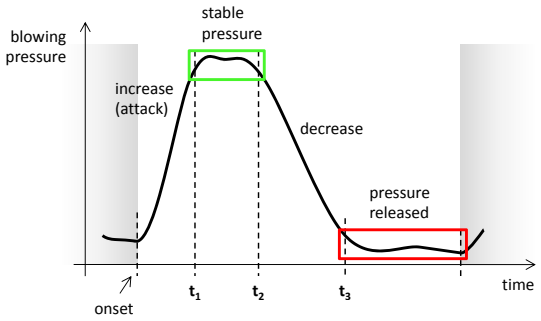


Figure 4: Schematic representation of the envelope model used in this work.

air pressure gets interrupted during note-to-note transitions (tonguing effect). In fact, one could interpret that in the staccato articulation notes are detached from each other by shortening the blowing pressure "pulses" and forcing a "silence" between consecutive notes. This makes clear an important difference between pressure profiles of staccato-like and legato-like note-to-note articulations.

The envelope model used for quantitatively represent blowing pressure profiles is depicted in Figure 4. It is based on dividing the pressure signal into four different segments. The first segment corresponds to the pressure attack and it is characteristic to all four articulation types. In a second phase (after t_1) the pressure reaches its maximum value. In all but the *full legato* articulation, t_1 defines the beginning of a stability state with a higher pressure, during which most of the energy is transferred to the instrument. The third segment, defined between t_2 and t_3 corresponds to a decrease of the blowing pressure, and its pressence is equally common to all articulations. Finally, the blowing pressure is released, and a state of stability at its minimum value is reached. In staccato articulations, the last state is significantly long and the pressure stays at 0 Pa. Conversely, the duration of this phase results extremely short for the case of legato articulations, mainly caused by the fact that blowing pressure is lowered and the air flow does not get completely interrumped.

The estimation of the segment durations (defined by t_1, t_2, and t_3) is carried out automatically for all notes in the database. The limits of each state are estimated by looking at how the instantaneous blowing pressure compares with a parameter $\Delta P = P_{max} - P_{min}$ that is computed for each note as the pressure dynamic range along its execution. The limits t_1 and t_2 are computed by considering that pressure excursion during the steady state segment must be within 90% of ΔP. Analogously, the time limit t_4 is defined by considering that the pressure excursion during the last state must be within 10% of ΔP. Once the profiles are segmented, durations and slopes are computed for each segment, with the idea of analysing the role of performance context parameters (dynamics, articulation, etc.) in shaping the envelopes of blowing pressure.

4.2 Observations on fingering and dynamics

A straighforward analysis was first carried out by looking at the averaged value of blowing pressure of the steady state segment (see Figure 4). For that purpose and with the aim of validating our findings in comparision with previous studies on the transverse flute [6, 3], computed pressure values were compared for different pitch values (finguerings) and articulations by averaging all corresponding notes in the database. The results are displayed in Figure 5, clearly showing how blowing pressure is related to pitch (fingering) and dynamics. The pitch-exponential nature of the relationships being independent upon the articulation used, corresponds with what had been shown in literature.

4.3 Attack times

By comparing the averaged attack time for each different articulation, an interesting observation can be made. For the case of *full legato*, in which the air flow is uninterrumped from note to note, the attack time appears as independent on the fingering (a similar behavior is observed for *legato* articulation). Differently, for those articulations in which the tonguing effect interrupts the blowing pressure right before the note onset, the attack time is negative-correlated to the pitch. Since the maximum blowing pressure before reaching a change of oscillation mode is in general lower for lower pitch fingerings, the performer risks entering an undesired second oscillation mode more easily. Thus, limiting the rate of increase of blowing pressure helps the performer to avoid entering in chaotic transitional states before reaching higher modes of oscillation. Within each type of these two articulation sub-groups, it remains clear that attack times are shorter for *legato* than for *full legato*, and also shorter for *soft staccato* than for *staccato*. Concentrating on one articulation type at a time, no significant differences were found when comparing the durations of the attack segments for different dynamics.

5. CONCLUSION

The main contribution of this paper is the acquisition and systematic analysis of blowing pressure signals from real performance in recorder playing. While previous studies had been mostly focused on the transverse flute, here we worked on the recorder and, most importantly, towards a parameterization of blowing pressure that would allow us the reconstruction of profiles with certain ease. With respect to the analysis of profile features, we have successfully reproduced some of the previous studies on transverse flute, and extended them by extracting and analysing articulation-specific features.

We will continue using the multi-modal database for approaching further challenges, like it is the case of building a generative model for synthesizing blowing pressure from an annotated score (possibly using more elaborate contour models (e.g. concatenated Bézier curves), studying mappings between blowing pressure and sound perceptual attributes, or driving physical models from recorded or synthetic blowing pressure signals.

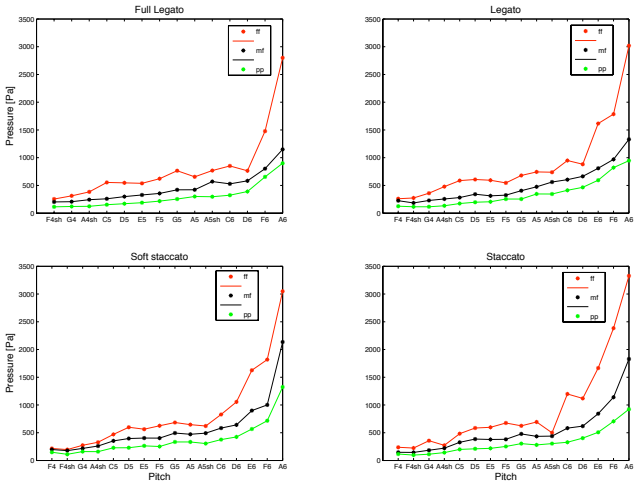


Figure 5: Average blowing pressure for different articulations versus pitch (fingering) for different articulations.

6. REFERENCES

[1] I. Cossette, P. Sliwinski, and P. Macklem. Respiratory parameters during professional flute playing. *Respiration physiology*, 121(1):33–44, 2000.

[2] P. de la Cuadra, B. Fabre, N. Montgermont, and L. De Ryck. Analysis of flute control parameters: A comparison between a novice and an experienced flautist. In *Forum Acusticum, Budapest*, 2005.

[3] N. Fletcher. Acoustical correlates of flute performance technique. *J. Acoust. Soc. Am*, 57(1), 1975.

[4] E. Maestre. *Modeling instrumental gestures: an analysis/synthesis framework for violin bowing*. PhD thesis, Universitat Pompeu Fabra, 2009.

[5] J. Martin. *The acoustics of the recorder*. Moeck, 1994.

[6] N. Montgermont, B. Fabre, and P. de La Cuadra. Flute control parameters: fundamental techniques overview. *ISMA*, 2007.

[7] A. Perez. *Enhancing Spectral Synthesis Techniques with Performance Gestures using the Violin as a Case Study*. PhD thesis, Universitat Pompeu Fabra, 2009.

[8] G. Scavone and A. da Silva. Frequency content of breath pressure and implications for use in control. page 96, 2005.

[9] M. Verge, B. Fabre, A. Hirschberg, and A. Wijnands. Sound production in recorderlike instruments. i. dimensionless amplitude of the internal acoustic field. *The Journal of the Acoustical Society of America*, 101:2914, 1997.

[10] M. Verge, A. Hirschberg, and R. Caussé. Sound production in recorderlike instruments. ii. a simulation model. *The Journal of the Acoustical Society of America*, 101:2925, 1997.

Experiences from video-controlled sound installations

Anders Friberg
Speech, Music and Hearing, KTH
Lindstedtsvägen 24
10044 Stockholm, Sweden
afriberg@kth.se

Anna Källblad
Studio 323 wip:sthlm
Strahlenbergsgatan 17
12145 Johanneshov, Sweden
anna.kallblad@bredband.net

ABSTRACT
This is an overview of the three installations Hoppsa Universum, CLOSE and Flying Carpet. They were all designed as choreographed sound and music installations controlled by the visitors movements. The perspective is from an artistic goal/vision intention in combination with the technical challenges and possibilities. All three installations were realized with video cameras in the ceiling registering the users' position or movement. The video analysis was then controlling different types of interactive software audio players. Different aspects like narrativity, user control, and technical limitations are discussed.

Keywords
Gestures, dance, choreography, music installation, interactive music.

1. INTRODUCTION
At the core of this work is the collaboration and flow between artistic and technical ideas and knowledge. In particular, we are interested in the relation between gestures and sound both in terms of the resulting sound and choreography. A classic example of such a collaboration is the New York 69th Regiment Armory in 1966 where 10 New York artists and choreographers worked with 30 engineers and scientists from Bell Telephone Laboratories to create groundbreaking performances that incorporated new technology [6]. Since then there have been a number of productions using dancers to control the music, i.e. [11][12].

Previously, we explored the possibility of using video cameras for analyzing gestures in conjunction with investigations of musicians' gestures as well as in previous interactive models including an interactive, collaborative game, Ghost in the cave [10] and a gesture controlled conducting system [5].

This is an overview of the authors' recent experiences with the three installations Hoppsa Universum, CLOSE, and Flying Carpet. We will here present both an artistic point-of-view, some technical methods and descriptions, as well as discussing the interaction between artistic goals and technical possibilities.

2. USER INTERACTION
The intention of these installations has been that the user will explore it without help or guiding. Also to let the user focus on her/his body and movements rather than to focus elsewhere, for example on a screen or device, which tends to steer away the perception of one's own body. There is also a conceptual artistic reason for this, namely the nature of the work in that it only exists when someone uses it. This might be the essence of dance: it only exists in the moment, nothing is left, only the experience and memory in the users/dancers' body. The purpose is the sharing and the experience in the moment, and the memory of that experience.

This self-exploration concept needs careful consideration in the technical design. Also, since this type of sensing does not involve any artifact or physical contact with an object, the natural ecological connection between sound source and object is broken. Although this is common in contemporary society with electronic devices it makes the interaction less intuitive. In particular in this case where the object is missing. One implication is that the response time of the system must be fast so that the user easily can associate a gesture with a certain change of the resulting sound.

There is a narrative created by the visitors own body movements in time. Through the interaction design one can suggest and/or control this narrative by facilitating certain choices of body movements and locations in the room. We were looking for an interaction that inspire movements that evoke certain feelings, and tell stories by how users movements develop individually and as a group. By changing the user interaction and the sound material over time (for example in a time cycle) a further enhancement of the narrative dimension is obtained. This will avoid the often static sound environment otherwise often resulting from installation work.

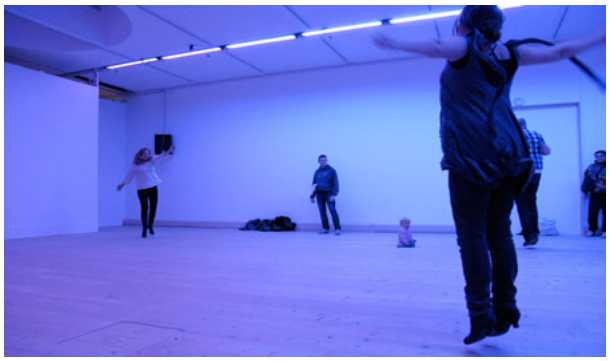


Figure 1. Hoppsa Universum during installation.

3. INSTALLATIONS

3.1 Hoppsa Universum
The artistic idea was to create a "magic room", where music and the light changed when you moved around. From looking at children's free movement, a choreography for five dancers was made for which the composer wrote and recorded the music. The music was then decomposed into five different scenes each with it is own interaction scheme. Each scene featured different trigger areas that enabled certain choreographed movement patters in the room. Both the sound and the light were controlled by the visitors movements. It was set up at Botkyrka Konsthall, see Figure 1.

The technical setup of Hoppsa consisted of 4 analogue video cameras in the ceiling covering the whole floor and 1

camera on the side for vertical sensing. All cameras were using infrared light. The length of each scene changed automatically providing a 20 min long narrative in total. A more detailed description is found in [9].

3.2 CLOSE – your body her voice

In this installation users trigger Palestinian women's recorded voices, see figure 2. CLOSE premiered 2009 at Sakakini Art Centre Ramallah and was then exhibited at Södra teatern Stockholm, Gottsunda dans och teater, Uppsala and Haninge konsthall 2010. We searched for a form of interaction that would evoke feelings of being limited, controlled and place users "involuntary" on intimate distance of each other in order to create a visual image of people bunched together. The process and the final artwork can be described as a physical interaction between ideas and context.

The background for CLOSE was the art project "Open Workshop for Culture and Arts" by ten Palestinian artists living in Gaza, Palestine and Israel with focus on the Palestinian women's public position from a feministic point of view, which resulted in a 20m long mural of which half stands in Ramallah and half in East Jerusalem. It shows women's full bodies in shapes inspired by literature references, an unusual sight in this context. Co-author and choreographer Källblad got the idea of a transient mural to connect live bodies and recorded voices. A mural is an ancient form of storytelling and usually through its construction fixed in one place and shape. The CLOSE mural would be the opposite. It's substance and meaning would appear when activated and stories would be retold in different orders and places depending on peoples movements.

In CLOSE the group had to cooperate and stay very close to each other in order to hear certain sounds. The trigger areas of these sounds were vaguely indicated in order to make people have the experience of searching and cooperating with strangers. One reviewer wrote "a practice in group dynamics" [3]. The language, Arabic, (translations were available) contributed to the context as well as available information of where the voices were recorded.

We used a laptop and webcams to facilitate travelling in restricted areas, thus the political situation dictated what technique we used. When travelling and collecting material we were daily affected by the occupation of the West Bank through roadblocks, shortage of water, questioning by soldiers etc. Long waiting for extra microphones due to difficulty in passing roadblocks etc all effected the working environment practically and emotionally. The feeling of constraint was overwhelming. We experienced how outside factors were filtered its way into the work. The outer context gave the form. Back in Stockholm building CLOSE we looked for an activation that would trigger the same emotion and image. The shape of the work giving its content, and vice versa.

The technical setup consisted of two webcams, laptop, 4 loudspeakers, and a carpet. The webcams were put in the ceiling making an approximate analysis of the number of people in each of 24 different zones using background subtraction (see below). A relative measure estimating the number of people in each zone was computed from the size of the blob, see Figure 4. When a visitor entered a zone, a single short sound was played. These sounds were short processed excerpts from the original recordings. When a group of visitors strolled along the carpet randomly, these sounds made a rather annoying and chaotic sound landscape. When the visitors cooperated and gathered in groups in a certain position, a single song or story was presented. This was determined by a threshold value of the estimated number of people.

Each group of visitors (10-30) was led into the exhibition area for 45 minutes. They were told that there was a carpet on which they could activate a soundscape where 15 single voices were hidden. To hear a single voice a number of people had to stand very close to each other. A guide was available to answer questions and could on demand hand out a map indicating where each voice was located. A typical action during the 45 minutes was that people first moved around quite fast and individually, then stepped out for a while to watch others, some then entered again and gradually started to make small groups in different areas and then finally forming one group that moved around with small steps and stopping to hear the single voices.

After each presentation the visitors could fill in a questionnaire with 15 questions consisting mostly of dichotomous Likert scales coded from 1 to 6. The overall impression (bad-good) was rated very positive with a mean of 5.5. Despite some comments relating to a frustration in the beginning they seem to have understand rather well how it worked and rated it (scale easy-hard) with a mean of 3.0. Three questions relating to cooperation issues and getting close to each other were all positively rated (being close: negative-positive, m=5.1; cooperation: difficult-easy, m=4.3, frustrating-rewarding, m=4.7). The visitors in Ramallah and at Södra teatern responded in a similar way with only small differences. Thus, in the end they got a positive experience of being close although there was some initial frustration in the beginning.

Figure 2. CLOSE at Södra teatern. A group of visitors are gathered in a group to hear a particular voice.

3.3 Flying Carpet

This is a dance/DJ installation that was commissioned for the Art's Birthday Party at Södra teatern, Stockholm, 2011. It was installed in a room next to a bar and was intended as a replacement for a typical dance floor in a club, see figure 3. In a club setting it is the DJ that is trying to play music that will enable people to dance. Here we turned that concept upside down, thus, it is the people that control the music instead, with the effect that the people had to make some effort to hear the music. It is an extra challenge to change such intuitive and well-known paradigms so we thought about how to help people to anyway be enabled to move more and attract more people to interact. Following quote was posted on the wall as instruction and inspiration:

Remember:

> "There are only two types of people in the world
> the ones that entertain and the ones that observe"
> Britney Spears

The video analysis was rather simple. Two webcams (Logitech Pro 9000) were placed in the ceiling above a carpet about 3.5X5 meter in size. Each camera extracted the quantity of motion (QoM, see below) and the centroid position of the motion in terms of x and y coordinates. The QoM value was controlling the overall state of the system divided into four

different interaction states using overlapping fuzzy functions (see i.e. [4]). Each camera controlled independently one channel of the resulting stereo audio output.

The first system state is when nobody is present on the carpet. Then a soft simulation of a wind sound is played.

The second state is when one person enters the carpet. This is a scratch mode using different samples about 10 s long taken from the record Movies for your ears by William Brunson [1]. The current sample is scrubbed (or scratched) with the x coordinate while the y coordinate is controlling the pitch. The sound is processed using a phase vocoder implemented by Ben Saylor and available in pd-extended version 0.41.4, [7].

The third state is the "disco" or dance mode. DJ Nico compiled a complete set consisting of three hours of electronic dance music. This music was continuously played but was not heard unless more people entered the carpet. The music was filtered with a bandpass filter using a variable q-value and centre frequency. When the QoM was relatively low the q-value was high resulting in a narrow band of music in which the filter frequency was controlled by the x-position. This resulted in a kind of phaser effect controlled by the movements. When QoM was relatively high the q-value was reduced resulting in the original full version of the music.

In the fourth state we attempted to further intensify the dance music by adding isolated voice samples taken from Movies For Your Ears. These were selected randomly and triggered on peak QoM values.

Considering the challenge of inspiring people to get involved we filtered the QoM signal so that its raise time was as short as possible and the decay time was fairly long, about 2 seconds. The fast response made the music easy to trigger and to start and the slow decay made it easier to maintain a high musical energy without too much physical effort. The intention was to do something similar to "spinning plates" tricks, thus, it was only necessary to make short "injections" of energy to get it running.

The informal feedback from the audience was very positive and many of the visitors that overcome the hesitation to try danced with much energy and big movements for a long time. Several people even suggested that we should apply for a patent.

Figure 3. A picture from the realization of Flying carpet at Södra teatern, Stockholm

4. TECHNICAL SETUP

4.1 Camera position and light

These installations were all made such that the users/visitors were free to move around and with a varying number of people. In this context, our preferred position is to place the video cameras in the ceiling pointing straight down. This makes it possible to register individual movements and to divide the floor in different areas, thus, facilitating using the room and space as part of the installation concept. It is obviously better the higher the camera is mounted since it reduces the differences in the video projection of people due to the view angle in the picture. Therefore the ceiling height is an important parameter and often a limitation when choosing installation space. It also affects the user experience. When the cameras are low the people recognize them fairly easily and can try to "cheat" by directing movement toward the cameras. If the cameras are located rather high the users might not understand how it is working, thus, creating a more magical experience.

The light will interact with the cameras if not infrared filters are used. Preferably, the light in the room should be either diffused i.e. evenly spread out like a normal office space or the light sources should be placed in the ceiling close to the cameras. In this way the influence of shadows is minimized.

4.2 Cameras

We used either analog cameras designed for surveillance (Hoppsa) or webcams. The analog setup has many advantages. It is well-known reliable technology that has been used for a long time. It has fast response time, demands little computational power and there is not any problem with cable length. Since the cameras were also designed to automatically switch to black and white night vision with led lights we used infrared filters so that we could change the light in the room without too much interference with the motion recognition. Thus, in the Hoppsa Universum installation we used 5 such cameras connected to two capture cards in one desktop Windows computer. All the processing including zone division and QoM computation as well as all audio computation with four-channel output was running on the same computer without problems.

For the other two installations we used a lighter setup with two webcams connected to a laptop. This makes the whole setup very portable and actually fits in a backpack. Active USB cables makes it possible to have cables with sufficient length (about 20 m). The cameras used (Logitech Pro 9000) have a zoom controlled by software. This facilitates the adjustment of the active sensing area during installation. However, using USB webcams with their software drivers can give unpleasant surprises. For example, it is often difficult to turn off the automatic light adjustments and sometimes this is not even possible (i.e. the built-in camera in Mac Book Pro). Cheaper webcams are not recommended since the background noise is often too high. The resolution of the camera is less important due to the inaccuracy in the analysis. Thus a common VGA resolution or smaller is sufficient. A third possibility is to use firewire for the camera communication and there are currently many small such semi-professional cameras available.

4.3 Video analysis methods

An overall design goal has been to keep the video analysis as simple as possible. Thus, only consolidated techniques have been utilized that are easy to understand, easy to interface and demands rather small computational power. We have used the EyesWeb program, v. 3, for all video analysis using existing analysis blocks, [2]. Two basic methods have been used for analyzing the visitors in the video picture.

Frame difference. The simplest and most robust technique is to compute the difference between consecutive frames. In this way all movement is registered but not a person standing still. A further improvement can be made by adding a second movement analysis with the black and white video inverted. From that initial extraction the total area is divided in different zones and for each one the quantity of motion (QoM) and the position is computed (centre of gravity for the motion blob). QoM is defined as the size of area of the frame difference after

some thresholding, see [2]. This was used in Hoppsa and Flying carpet.

Background subtraction. An alternative method is to subtract a still picture of the background from the incoming video. In this way the whole area of each person is visible independent of if they move or not. This was used in CLOSE, see Figure 4. It feasible to use in an installation even during a longer time but it is more sensitive to changes in light and has to be more carefully calibrated.

Obviously, using these techniques it will only work if there is a difference in brightness between the user and the background. Usually, a light carpet color is in many cases better than a dark one. Often people (at least Swedish art-oriented ones) tend to have dark or often only black clothes. Alternatively the carpet could have a pattern of both dark and light colors. This would make the analysis independent of clothing or skin color.

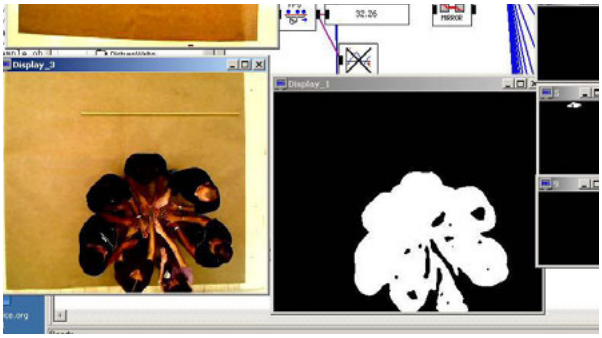


Figure 4. A screenshot of the video analysis during the testing of CLOSE. Left display is the raw video and the right display shows the detected area in white.

5. DISCUSSION

It has been challenging to find workarounds for the artistic ideas due to the technical constraints. However, these constrains can also be considered in the same way as budget and time schedule limitations - what is possible to do within these parameters? Then one has to try to be as clever as possible and to find a form that is interesting in itself so that the constraints serves as and can be used for artistic purposes. There is also joy from "beating the odds" that feeds energy to a project, as well as when artistic ideas can spur technical solutions.

Was the technical level enough for realizing the artistic goal? The answer depends partly on how open-ended the artist intended the work to be. Due to the self-exploratory design of the installations, the coupling between gesture and sound had to be immediate and intuitive. This made it necessary to constrain the interaction to rather simple models. Our experience is that even very simple actions like triggering a sound on a certain position is not grasped by every user, in particular, if there are several users active at the same time. In this view it is hard to see that more advanced video analysis would substantially improve the user experience. For example, in Flying Carpet even a simple division of QoM can make the system behave in four different states depending on the number of active users. There are from an artistic point of view pros and cons to the users' fully understanding how the triggering works, a certain "mystique" can be intriguing.

We found that the fine-tuning of the different parameters was a crucial step that needs to be carefully considered both during testing and development and, in particular, at the final location, considering both the type of room and audience.

Overall the visitors seemed quite willing to lend their own bodies to fill or activate these installations. Their level of awareness or opinion that they themselves by this activity in fact make the artwork or performance remains unsaid.

A different approach is to use the video gesture analysis material directly and after further video processing project it as a part of the background or scenography. This has been explored by Frieder Weiss in a number of applications including dance performances [8]. An interesting future extension would be to combine the music interaction with light control using video processing and data projectors.

6. ACKNOWLEDGEMENTS

Hoppsa Universum by Anna Källblad in collaboration with Niko Rölke, music; Karl Svensson, light design; Anders Friberg, audio-video interaction; Tove Axelsson, set design; Linda Adami, Kerstin Abrahamsson, Johanna Klint, Maryam Nikandish, Stina Nyberg, dance. Supported by Swedish Arts Council, City of Stockholm, Stockholm County Council, City of Botkyrka, University of Dance and Circus, and Moderna dansteatern.

CLOSE by Anna Källblad and Annette Taranto in collaboration with Amal, Asmaa, Oraib, Rasha, Rula, Ruba, Majdal, Juma, Nadia, Zeynab, Leila, Kamelia, Mohah, Fatima, Rowan, El-Funoun Dance Troupe, voice; Jaime Fawcus sound-engineering, Anders Friberg audio-video interaction; Karl Svensson light design; Mervi Junkkonen, documentation. Supported by Swedish Arts Council, The Swedish Arts Grants Committee, and The Swedish Institute, City of Stockholm.

Flying Carpet by Anders Friberg and Anna Källblad in collaboration with Bill Brunson, music samples, DJ Nico, music compilation. Commissioned by Södra Teatern.

7. REFERENCES

[1] Brunson, B. Movies for your ears, Electron Records, 2004.

[2] Camurri A., Mazzarino B, Volpe G. Analysis of Expressive Gesture: The EyesWeb Expressive Gesture Processing Library. In Gesture-based Communication in Human-Computer Interaction, LNAI 2915, Springer Verlag, 2004.

[3] Claeson, A. Uppsala Nya Tidning, May 6, 2010.

[4] Friberg, A. A fuzzy analyzer of emotional expression in music performance and body motion. Proc. of Music and Music Science, Stockholm 2004, 2005.

[5] Friberg, A. Home conducting: Control the overall musical expression with gestures. Proc. International Computer Music Conference, (pp. 479-482), 2005.

[6] http://www.9evenings.org

[7] http://puredata.info/downloads

[8] http://www.frieder-weiss.de/

[9] Källblad, A., Friberg, A., Svensson, K., & Sjöstedt Edelholm, E. Hoppsa Universum – An interactive dance installation for children. Proc. of New Interfaces for Musical Expression - NIME, Genova, 2008.

[10] Rinman, M-L., Friberg, A., Bendiksen, B., Cirotteau, D., Dahl, S., Kjellmo, I., Mazzarino, B., & Camurri, A. Ghost in the Cave - an interactive collaborative game using non-verbal communication. In Gesture-based Communication in Human-Computer Interaction (pp. 549-556), Springer Verlag, Berlin, 2004.

[11] Siegel, W. and Jacobsen, J. The Challenges of Interactive Dance: An Overview and Case Study. Computer Music Journal, 22(4), 29-43, 1998.

[12] Winkler, T. Motion-sensing music: Artistic and technical challenges in two works for dance. Proceedings of the International Computer Music Conference, 1998.

ROOM#81 - Agent-Based Instrument for Experiencing Architectural and Vocal Cues

Nicolas d'Alessandro
MAGIC Lab, University of British Columbia
Vancouver, BC, Canada
nda@magic.ubc.ca

Roberto Calderon
MAGIC Lab, University of British Columbia
Vancouver, BC, Canada
rvca@interchange.ubc.ca

Stefanie Müller
Hasso Plattner Institute
Potsdam, Germany
stefanie.mueller@student.hpi.uni-potsdam.de

ABSTRACT

ROOM#81 is a digital art installation which explores how visitors can interact with architectural and vocal cues to intimately collaborate. The main space is split into two distinct areas separated by a *soft wall*, i.e. a large piece of fabric tensed vertically. Movement within these spaces and interaction with the soft wall is captured by various kinds of sensors. People's activity is constantly used by an agent in order to predict their actions. Machine learning is then achieved by such agent to incrementally modify the nature of light in the room and some laryngeal aspects of synthesized vocal spasms. The combination of people closely collaborating together, light changes and vocal responses creates an intimate experience of touch, space and sound.

Keywords

Installation, instrument, architecture, interactive fabric, motion, light, voice synthesis, agent, collaboration.

1. CONCEPTS AND OUTLINE

Human social interaction is versatile and pervasive in human life. Yet, when creating machines for interaction we often forget the subtlety of unconscious cues and focus on conscious models. Looking at intimate inter-personal relations between humans, we can state that these unconscious parts – such as small gestures or digressing eyes – lay out the foundation for emotional commitment. We propose that human-computer interaction needs to be defined through both conscious and unconscious interactions that rely on meaningful feedback systems.

Although art has made use of human-computer interaction through *new media installations*, it generally focuses on conscious and direct interaction paradigms, like the feedback loop, to create a simplistic illusion of control. We believe that there is a need to explore the integration of such uncontrollable and, at times, ungraspable nature of unconscious interactions between humans and their machines. As a result, the dialogue between humans and their technologies remains personal and intimate.

In ROOM#81 we examine interactive places that explore interaction through subtle contexts. Visitors are welcomed in a room where architectural and vocal cues are the main components that structure the nature of such space. A large piece of fabric is hung up in the middle of the room to create a *soft wall* separating the space in two areas. Visitors, who have never seen each other before, can access the installation from both sides of the fabric and have an interaction between themselves by pulling and pushing the fabric. Their movements in space, along with their haptic interaction with the fabric affect the sentient nature of the room, which responds with changes in light and voice modulations. Visitors experience an invisible, yet personal, vocal character that screams in agony, pleasure, or concern somewhere in the room they intimately share.

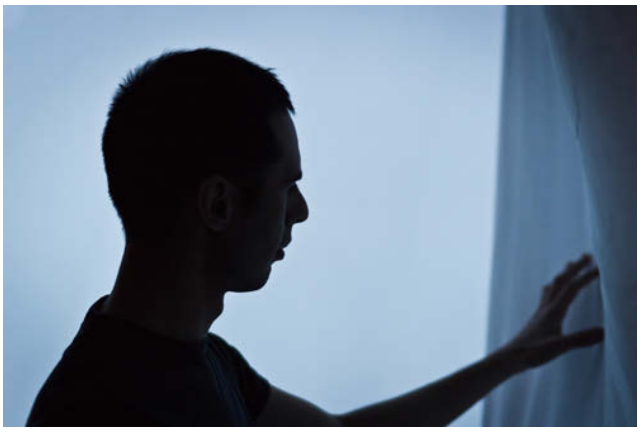


Figure 1: Movements in the installation space and collaborative interaction with the fabric influence the lighting properties and the voice quality of an artificial vocal character, screaming in the room.

The soft wall: a mediation tool for intimacy

We believe that these three simple cues – the movement of a foreign person towards you through a piece of fabric, the changes in light quality and the changes in the tension of a voice – open up a large space for aesthetic interpretation. Based on intimate and sensual displacements of the fabric, one begins to wonder the nature and story of the person behind it. How does this person look like? It is a man or a woman? What is his/her personality? Where does he or she come from? These questions make the *soft wall* both an invitation to play and a collaborative effort to affect the nature of the room. Figure 2 shows a close-up on interlaced hand gestures of people that have never seen each other.

The vocal character: a subjective response

The large spectrum of vocal solicitations adds a second layer to our exploration. We can easily imagine some visitors being amazed by the sharp and quick spasms of the voice synthesizer. Yet, we also think of visitors who might regard

Figure 2: Confident interaction from both sides of the fabric with people who do not know each other, making the fabric a mediation tool for intimacy.

such sounds as painful screams, or clear sexual references. The same is true for slow and languorous sounds or whispering noises. Because people react differently to the same kinds of vocal stimuli, our array of stimuli becomes infinitely large. ROOM#81 positions itself at the subjective level of human interaction, opening a wide space for interpretation.

The agent: analog instrument and social control

With this collaborative instrument we also offer a greater social sense of control. Visitor's interaction is never directly mapped onto the vocal or light spaces, but has a non-obvious, adaptive impact on the vocal and light stimuli. Gestural inputs from sensors are used in an ongoing machine learning process that constantly changes the behaviour of an agent. As the agent forms a model of its world and acts upon it, its "thoughts" are mapped on the vocal and light spaces. With this in mind, a visitor can only control the other person's reaction to his own usage of the fabric. As such, we conceive ROOM#81 as an analog instrument of a tripartite nature, that is, played by two humans that intimately discover each other through a *soft wall* and an architectural/vocal agent.

2. RELATED WORK

ROOM#81 is related to interactive art for connecting people through sound haptics, voice-related digital musical instruments and interactive light/sound installations.

Connecting people through sound

Contacts [6] is an interactive sound installation for two or more people that consists of a small ball on a stand. When the visitor places his or her hand on the ball, his body becomes an interactive sound space that is sensitive to other people's touch. Shaking hands, caressing and kissing create different sonorous sounds. If the visitor remains alone, nothing happens. Following this, the visitor is encouraged to explore the intimate space of touch with a second person. To detect touch, the installation makes use of the small electrical tension every human carries on his skin surface.

Akousmaflore [5] is based on the same concept as Contacts. In Akousmaflore the visitor strokes musical plants that are arranged in a small garden. Each plant reacts in a different way to contact and warmth based on its individual leave structure. Visitors can interact with each other by using the plants as different instruments in their musical arrangements.

Both installations aim at creating a complex and subtle interaction between gestures and sound. However, they often implement random factors on their process of sound mapping through various modifications on the data. Furthermore, both have used digital audio samples in order to create harmonic feedback.

Voice-related digital musical instruments

The use of interactive voice synthesis for both performative and installation purposes has not been studied further than sample playback. We can highlight HANDKSKETCH [2] and DIVA [3] as two more advanced projects.

The HANDKSKETCH [2] is a digital instrument made for the bi-manual control of voice quality dimensions: pitch, intensity and glottal flow parameters. The instrument aims at exploring the expressivity of voice by simulating laryngeal behaviours with realtime gestural control.

DIVA [3] allows for direct manipulation of phonetical and prosodical spaces by using hand gestures. Hand gestures are intermediately converted to articulator (e.g., tongue, jaw, lip, vocal chords) parameters of a 3D vocal tract model.

Interactive light and sound installations

Shortcut [1] is an artwork that responds to the speed, rhythm and number of people in a passageway by building up a pattern of light that reflects the recent movements. A similar concept is deployed by Dune [4] which maps a sound and light space onto a visitor's movements through space.

3. INSTALLATION SETUP

In this section, we explain how the installation is laid out. We also give further details on the inputs (webcam, stretch sensor and light sensors) and outputs of the system (two-channel audio system, beamer lighting).

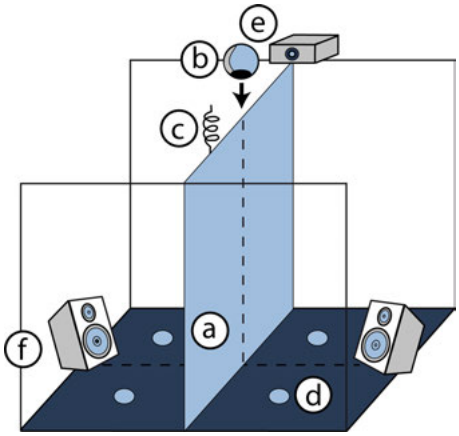


Figure 3: Setup of ROOM#81: (a) large piece of fabric, (b) webcam, (c) stretch sensor, (d) four light sensors, (e) beamer, (f) two loudspeakers.

Spacial arrangement

ROOM#81 is an installation that is setup on a rectangular floor space of 4x4 meter. At least one side of this space requires a wall from the building. Colour and material of this wall can be of any kind, but light colours are preferred. In Figure 3, this required pre-existing wall is the vertical face of the cube that is closest to the viewpoint.

Collaborative piece of fabric

Perpendicular to this wall, we place a piece of fabric (a). The fabric divides the room into two parts of equal size. The width of the fabric is at least 4 meter to give visitors a surface large enough to discover different kinds of interaction. The height of the fabric is at least 2 meter to avoid

that visitors look over it to the other side of the installation. In our current prototype of ROOM#81, we hang up the fabric between two solid tripod stands (people push and pull the fabric). Visitors can access the fabric from both sides ideally by two separate entrances. A more open configuration is possible, but the notion of two distinct area should be maintained.

Webcam, stretch sensor, light sensors

A webcam (b) is placed 1 meter above the fabric to capture the actions of visitors on both sides of the installation. In our setup, the webcam is attached to the ceiling of the room. However, if the ceiling is too high or not appropriate for positioning a webcam, it can also be placed on the side of the fabric to track lateral movements. Figure 4 shows results of the webcam-based motion tracking.

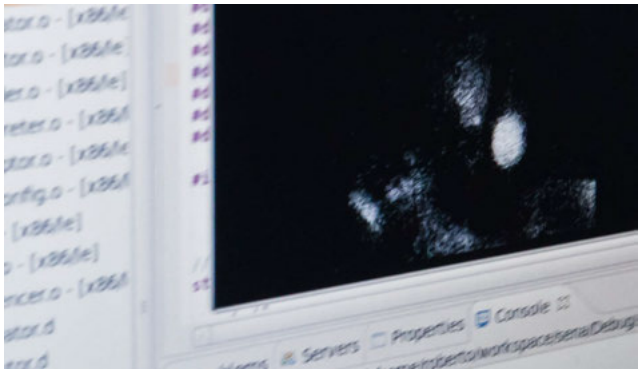


Figure 4: Results of the webcam-based motion tracking inside the installation space (left side).

A stretch sensor (c) is attached on top of the fabric. It measures the degree of tension in the fabric. The stretch sensor enables us to reason if visitors applied soft or hard manipulation when pulling or pushing the piece. In our current prototype, we horizontally hang up a metal bar between the two tripod stands to fix the stretch sensor.

Four light sensors (d) are placed on the ground, one in each quarter of the installation space. The light sensors capture additional information on visitors actions in the installation. People's movements relatively to the light source create various shades that are captured by these sensors. The captured data provides a rough sense of where people are located in the room.

Ambient light and vocal sound diffusion

We place a video projector (e) at the top corner of the fabric that is not connected to the wall. We use the video projector to vary the ambient light in the installation by going over a range of single colours that are projected full-screen on the wall. By projecting different colours on each half of the wall, we are able to create different moods on each of the two sides of the installation.

One loudspeaker (f) on each side of the installation diffuses the vocal sounds, avoiding the creation of an immersive sound field. This localizes the voice on various sources across space. Consequently, the virtual vocal character moves from one area to the other depending on visitors' behaviour.

Second wall: closing the space

Optionally, a second wall can be placed on the opposite of the pre-existing wall. In Figure 3, this second wall is the vertical face of the cube that is furthest to your viewpoint. Using a second wall helps to close the space which gives visitors a sense of a semi-private surrounding. Visitors interact in a less constrained manner with such an arrangement.

4. THE AGENT BEHAVIOUR

In this section, we explain how we make use of a constantly learning agent impacting on architectural and vocal cues, in order to encourage the visitor to reflect upon his behaviour. We describe how this machine learning process works by using a Bayesian network. Afterwards we give more details on the voice synthesis algorithm for the production of spasms.

Agent-based interaction

There is an interesting space to explore between the instrument and the installation. For an instrument we expect a predictable behaviour that allows for practice. Opposed to this is the installation which introduces unpredictable aspects from the visitor's point of view. Indeed the visitor embodies a part of the system, but is also partially embodied by the system. In this context, the use of an agent as another contributor to the experience is particularly suitable. Participants can enjoy and refine their use of a predictable part of the installation – the two-sided fabric – but their movements are used to train an agent that takes part in the experience by influencing light and sound.

Self-learning Bayesian network

The inputs of the system are the image captured by the webcam, the stretch sensor and the light sensors. Such data is fed into a Bayesian network that aims at predicting human behaviour in the installation. As visitors interact with the installation, data is created and the installation becomes more accurate. An agent then uses the network in a statistical manner to predict visitors' behaviours or promote them. Our installation makes use of no predefined mapping, but uses adaptive machine learning to create the visual and audible cues. Therefore, our instrument is self-determined and self-learned based on the visitor's interaction.

Visitors contribute to the ongoing learning process of the agent which allows for complex scenarios. For example, if a visitor pulls hard over a long time, the room will not necessarily stop screaming after the visitor disrupted his interaction. Based on the learning algorithm, ROOM#81 will behave differently for each visit, sometimes extrapolating the solicitation, sometimes provoking the change.

Voice synthesis algorithm

The agent shares his beliefs (statistical probabilities of certain actions) with an interactive voice synthesizer. The voice synthesis algorithm is based on the RAMCESS synthesis engine, the same as used in the HANDSKETCH digital instrument. RAMCESS is a concatenative synthesis (using FTM for Max/MSP) with realtime frame selection and sound transformation. This algorithm produces primitive vocal spasms – like a big open /a/ – with realtime control on the pitch, intensity, vocal fold tenseness and breathiness.

The synthesis parameters are not controlled directly but are mapped to a vocal space. The agent changes the way the system cycles through this vocal space. There are three dimensions in how this cycle changes based on agent probabilities: the speed of the cycle which determines the abruptness of the spasm, the overall pitch zone of the spasm which can be low or high, and the overall pitch range of the spasm which can be flat or abrupt. The sound is also spatialized between two loudspeakers, in order to set the voice where there is the least chance that something new happens, as a way of triggering a too predictable visitor.

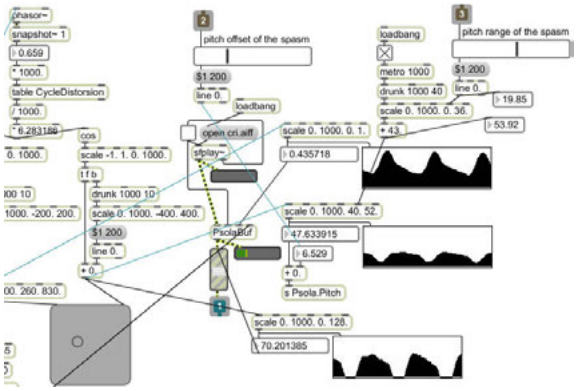


Figure 5: Voice synthesis mapping in Max/MSP. Trajectories in the vocal space can be changed: spasm, speech pitch offset and pitch range.

5. EQUIPMENT

ROOM#81 requires the following equipment: 1 piece of fabric: size = 4×2 meter; 2 tripods: height = 2 meter; 1 metal bar: length = 4 meter; 1 webcam; 1 beamer; 1 stretch sensor; 4 light sensors; 2 loudspeakers; elements to put vertically and create a second wall, e.g. poster grids work (optional). We can bring all of the equipment except for the tripods, the loudspeakers, the metal bar and the optional second wall. We would like to ask conference organizers to provide us with these items. There are no more technical requirements except for a free white wall. ROOM#81 can be set up as both a foyer location or a room-based installation.

6. VIDEO DEMONSTRATION

A video of the installation has been taped during prototyping at the University of British Columbia (Vancouver):
`http://www.nicolasdalessandro.net/room81`

7. ACKNOWLEDGEMENTS

The authors would like to thank the Media and Graphics Interdisciplinary Centre (MAGIC) of the University of British Columbia, and its Director: Prof. Dr. Sidney Fels. First because this lab has created the opportunity for us to meet and discuss about connecting our respective matters of interest. Secondly because we could use MAGIC facilities to mount the prototype and develop the software.

8. BIOGRAPHIES

Nicolas d'Alessandro

Nicolas d'Alessandro is a researcher and musician who has been exploring the interactive side of artificial voice production for the last eight years. He built several digital instruments for performing synthetic speech and singing, such as the HANDSKETCH, and played them on stage. He holds a PhD in Applied Sciences from the University of Mons (Belgium) and is now Research Associate at the University of British Columbia (Canada), where he supervises the DIVA project and co-directs the UBC Laptop Orchestra.

Roberto Calderon

Roberto Calderon is an architect and artist interested in in the human perception and interaction with ubiquitous technology and interactive environments. His work deals with public displays, interactive architecture, wearable and mobile devices. He is interested in the concept of agent based architecture able to form intimate relationships with its inhabitants. He is currently pursuing his PhD at the Media and Graphics Interdisciplinary Centre at the University of British Columbia.

Stefanie Müller

Stefanie Müller is a computer scientist and author interested in transferring the story behind everyday experiences into interactive artwork. Thereby, she is drawing on her experiences as a writer of modern poetry for which she received several scholarships. Stefanie is working as an anthologist and recently published two books in collaboration with the canadian photographer Darren Holmes.

9. REFERENCES

[1] J. Bruges. Shortcut. `http://www.jasonbruges.com/projects/uk-projects/shortcut`, 2010.

[2] N. d'Alessandro and T. Dutoit. HandSketch Bi-Manual Controller: Investigation on Expressive Control Issues of an Augmented Tablet. In *New Interfaces for Musical Expression*, pages 78–81, 2007.

[3] S. Fels, R. Pritchard, and A. Lenters. Fortouch: a wearable digital ventriloquized actor. In *New Interfaces for Musical Expression*, pages 274–275, 2009.

[4] D. Roosegaarde. Dune. `http://www.studioroosegaarde.net/project/Dune`, 2010.

[5] Scenocosme. Akousmaflore. `http://www.scenocosme.com/akousmaflore_en.htm`, 2009.

[6] Scenocosme. Contacts. `http://www.scenocosme.com/contacts_installation_en.htm`, 2009.

Kinetic Particles Synthesizer Using Multi-Touch Screen Interface of Mobile Devices

Yasuo Kuhara Daiki Kobayashi
Department of Interactive Media, Tokyo Polytechnic University
1583 Iiyama Atsugi Kanagawa Japan 243-0297
kuha@t-kougei.ac.jp m0724044@st.t-kougei.ac.jp

ABSTRACT
We developed a kinetic particles synthesizer for mobile devices having a multi-touch screen such as a tablet PC and a smart phone. This synthesizer generates music based on the kinetics of particles under a two-dimensional physics engine. The particles move in the screen to synthesize sounds according to their own physical properties, which are shape, size, mass, linear and angular velocity, friction, restitution, etc. If a particle collides with others, a percussive sound is generated. A player can play music by the simple operation of touching or dragging on the screen of the device. Using a three-axis acceleration sensor, a player can perform music by shuffling or tilting the device. Each particle sounds just a simple tone. However, a large amount of various particles play attractive music by aggregating their sounds. This concept has been inspired by natural sounds made from an assembly of simple components, for example, rustling leaves or falling rain. For a novice who has no experience of playing a musical instrument, it is easy to learn how to play instantly and enjoy performing music with intuitive operation. Our system is used for musical instruments for interactive music entertainment.

Keywords
Particle, Tablet PC, iPhone, iPod touch, iPad, Smart phone, Kinetics, Touch screen, Physics engine.

1. INTRODUCTION
Various musical video games such as Namco's Taiko Drum Master, Konami's Guitar Freaks, and Nintendo's Wii Music have been developed because of the requirement to play music with instruments. However, with these games, a player cannot perform his/her own music, but passively operates the controllers according to preloaded music. On the other hand, some portable synthesizers such as Korg's Kaossilator and Nintendo DS's DS-10 and M01 have been developed for musical performance. Also, Yamaha developed Tenori-on as a music pad that creates music visibly by a finger touching interface. However, it takes a long time to learn how to play music on these portable synthesizers because of the difficult interfaces. Therefore, there are many precedents [1] in the NIME community related to investigating controllers oriented towards musical experience. Furthermore, various smart phones such as iPhone and Android devices are in widespread use, and are highly evolved as application devices beyond simple mobile phones. The new field of mobile music emerged at the intersection of ubiquitous computing and portable technology for a musical expression [2]. Many musical applications for the smart phone such as MoMu [3] have been developed and distributed on the web site.

In this paper, we proposed a simple sound generator using multi-touch gestures by fingers, which is familiar to those who have experience of operating popular smart phones or mobile PC devices. Most of them have a three-axis acceleration sensor, which enables the user to operate by tilting the device. Accordingly, they are useful for a novice for composing and performing music easily. We aimed to develop a mobile synthesizer suitable for musical performance with a touch screen that is able to generate attractive music while keeping the operation simple, which makes it possible for everyone to enjoy performing music.

We used a moving particles model as a sound generating unit. Many sounds in the natural world are made from an assembly of simple sound components, for example, rustling leaves, falling rain, babbling streams, ocean surf, forest sounds, and crowd applause. We regard a particle as such a sound component, which is the source of sound synthesizing. Each particle moves in the screen via the kinetics, sounding by synthesizing its own physical properties. A player can operate particles by touching the screen and tilting the device, and totally all of the particles generate music sounds.

2. METHOD
2.1 System Configuration
Our system is on two platforms: tablet PC and iPhone/iPad with a touch screen. For PC, we use Windows 7, and FlashDevelop software [4], which is a free ActionScript source code editor that generates Flash movies. For iPhone/iPad, we use Apple computer's iOS SDK. In both of platforms, we use Box2D [5], which is a free physics engine that can calculate the physical dynamics of a large number of particles. The iPhone/iPad has a three-axis acceleration sensor and a player can move particles by tilting the device.

2.2 Sound Synthesis
Our system synthesizes sounds using the parameters related to the motion of particles in the screen, which move in accordance with the law of kinetics. When a player touches a screen, a particle is generated in the touched place of the screen and begins to move. The particle is continuously moving and frequently collides with other particles or some stable walls initially located in the screen. The kinetic motion and collision cause the sound synthesis. Consequently, our system composes music of moving particles in the screen. Some demonstration movies are shown on our website [6].

2.2.1 Kinetic motion

A particle works as an oscillator of an analogue synthesizer, which generates tone by self vibration such as sine wave, square wave, and saw wave. Particles move and spin by their kinetic properties such as linear and angular velocity. When a particle is generated by the player touching the screen, it has their initial value.

Each kinetic parameter is related to a sound synthesis element (see Table 1 and Figure 1). Linear velocity is used for the amplitude of tone, and angular velocity is used for pitch, which is the frequency of tone. The particle shape is related to the wave of the oscillator, for example, a circle is for a sine wave, and a rectangle is for a square wave.

Table 1. Particle kinetics and sound synthesis

Particle dynamics	Sound synthesis
Linear velocity	Amplitude
Angular velocity	Pitch, Frequency
Shape	Oscillator waveform
Collision	Percussive sound

2.2.2 Collision

When a particle collides with others or wall objects, a percussive sound is generated. Each particle has its own percussive sound of collision assigned in advance. When a player shuffles the device, it acts like maracas, because many particles make sounds by the collision effect.

The friction and restitution are related to the activity of the motion of particles. At the point of the contact, the friction decreases the kinetic energy. If a player sets a larger friction parameter, it is easier for the motion of particles to be inactive. As a result, the music becomes quiet. In contrast, the restitution provokes the rebound to a particle at the collision, which increases the kinetic energy. Consequently, the motion of particles becomes more active and the music is more aggressive.

2.3 Performance

At first, there is no particle in the screen, except preset walls, which are optional. A player performs music by touching the screen to generate moving particles. If the number of particles is few, the music is quiet. On the other hand, a player can generate one particle after another by continuous streams of touches on the screen. A large amount of particles, which means many oscillators in the screens, makes the music loud. If a player touches the particle by finger, the touched particle is erased, while touching a void space in the screen causes generation of a new particle. A player can adjust the loudness of music by controlling the number of particles.

In our natural environment on Earth, all objects are influenced by G-forces as acceleration of gravity. When a player tilts the device, particles are forced to increase the velocity to the tilted direction by the working of the three dimensional accelerometer. As a result, motion or collision of particles is promoted, and it causes the sound to be more active. If a player shuffles the device, the motion of particles is more active, and the music may become more aggressive.

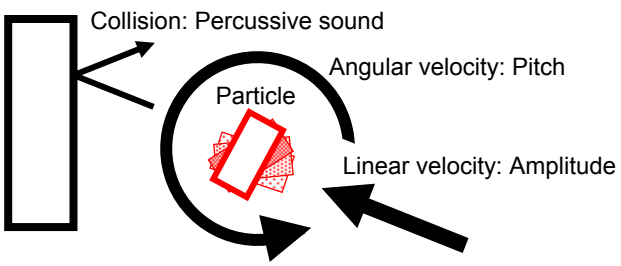


Figure 1. Particle kinetics and sound synthesis.

2.4 Graphics

A particle is drawn with the frame color according to its kinetic energy calculated from linear and angular velocity. The order of coloring graduation corresponds to the hue cited sequence, which is similar to Newton's sevenfold, gradually changing from red, orange, yellow, green, blue, indigo and to violet. Additionally, some textures are able to be attached to the face of the particle like patterns of tops. Players can enjoy a visual variety of particles moving and coloring while they perform music (see Figure 2).

3. DISCUSSION

This system works as both an analogue synthesizer and percussive instrument. A player can adjust the balance of these two musical aspects. If a larger collision sound is set, a player can perform rhythmical music like using percussion instruments. By setting stable walls or fences, more varied collision and motion occur, which promotes musical and visual attraction.

On the other hand, if a large oscillator sound is set, a player can perform the ambient music of an analogue synthesizer. By a large amount of particles, which are oscillators, various frequency waves are mixed and complex sounds are generated.

This system has a simple operating interface of touching and tilting the device. It enables everyone to perform music. In the demonstration, children enjoyed this system as a musical toy by touching the screen and shuffling the device.

4. CONCLUSION

We developed a simple analogue synthesizer and percussive instrumental sequencer using the multi-touch screen interface of a mobile device. Using this system, people who do not have musical skill can perform interesting music. In the near future, we will improve this system for playing more varied instruments and enjoying the ensemble of sounds and images, while keeping the operation of the multi-touch screen simple.

5. REFERENCES

[1] Blaine, T. The Convergence of Alternate Controllers and Musical Interfaces in Interactive Entertainment. In Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05), Vancouver, Canada. 2005, 27-33.

[2] Gaye, L. Holmquist, L. E. Behrendt, F. and Tanaka, A. Mobile Music Technology: Report on an Emerging Community. In Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06), Paris, France. 2006, 22-25.

[3] http://momu.stanford.edu/

[4] http://flashdevelop.jp/

[5] http://www.box2d.org/

[6] http://www.media.t-kougei.ac.jp/kinepati/

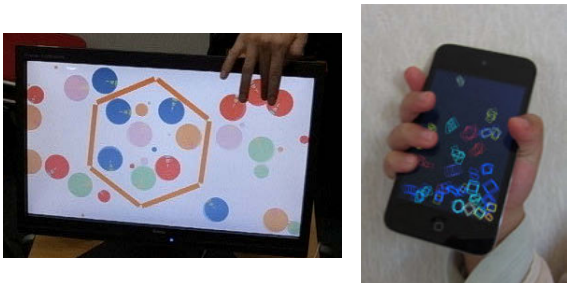


Figure 2. Example of kinetic particles synthesizer.

The Sound Flinger: A Haptic Spatializer

Chris Carlson
CCRMA - Stanford University
660 Lomita Dr.
Stanford, CA 94305
carlsonc@ccrma.stanford.edu

Eli Marschner
CS - Stanford University
353 Serra Mall
Stanford, CA 94305
eli@cs.stanford.edu

Hunter McCurry
CCRMA - Stanford University
660 Lomita Dr.
Stanford, CA 94305
phunter@ccrma.stanford.edu

ABSTRACT
The Sound Flinger is an interactive sound spatialization instrument that allows users to touch and move sound. Users record audio loops from an mp3 player or other external source. By manipulating four motorized faders, users can control the locations of two virtual "sound objects" around a circle corresponding to the perimeter of a quadraphonic sound field. Physical models that simulate a spring-like interaction between each fader and the virtual sound objects generate haptic and aural feedback, allowing users to literally touch, wiggle, and fling sound around the room.

Keywords
NIME, CCRMA, haptics, force feedback, sound spatialization, multi-channel audio, linux audio, jack, Arduino, BeagleBoard, Pure Data (Pd), Satellite CCRMA

1. MOTIVATION AND DESIGN
The Sound Flinger is an interactive sound spatialization instrument that allows users to touch, position, and throw sounds around a physical space. Our goal was to create a device that encourages playful experimentation and is approachable for uninitiated users while being complex enough to allow development of more advanced, if not virtuosic, techniques.

The instrument is situated in the center of a quadraphonic sound field. Users may grab and move up to four sliders that are positioned in a square configuration. If a user positions a slider at the current location of one of two virtual sound masses a force pulling the slider toward the mass will be felt. A pitch-based modulation of the sound associated with the mass occurs as the attraction between the slider and the mass increases, providing additional auditory feedback. Once a slider "latches" onto a sound mass users may wiggle the mass back and forth, or fling it toward another slider.

New sounds may be recorded into one or both virtual sound masses by connecting an external audio source and pressing one or both record buttons on the surface of the instrument.

2. HARDWARE
The heart of the Sound Flinger architecture is an embedded programming platform called Satellite CCRMA [8]. This platform consists of a Texas Instruments BeagleBoard [2] running a Linux distribution with Planet CCRMA audio packages [4] and other open source software. The integration of this platform allows the Sound Flinger to operate without being tethered to a laptop. The only external connections on the instrument are a 12V DC power supply, a 1/8" audio line in, and four 1/4" audio outputs to a mixer.

The primary hardware components include four motorized linear potentiometers (part no. ALPS RSA0N11M9A05) arranged along the edges of the instrument's square enclosure. Two momentary push buttons are placed at opposite corners. The sliders and buttons are embedded in a 9" × 9" sheet of Plexiglas on top of a wooden box containing an ATmega328-based Arduino Nano [1], an ARM-based BeagleBoard, a combination USB hub & Ethernet adapter (for remote programming), two AVR dual H-bridge motor controllers [5], a SIIG IC-710112-S1 USB Soundwave 7.1 Digital audio interface, and connective circuitry on a breadboard.

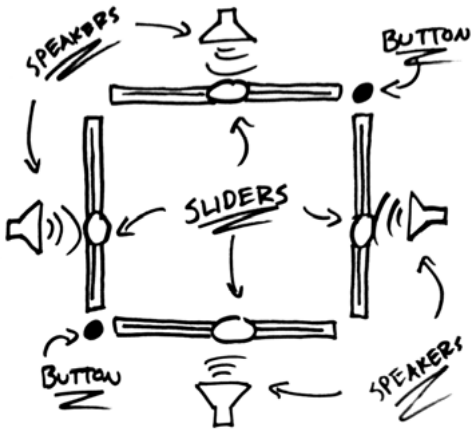


Figure 1. Control Design Concept

The instrument is positioned such that each attached speaker lines up with the center of a side, which corresponds to the center of a slider, as shown in Figure 1. All external connections are located on the bottom of the box to allow external wiring to be hidden. A 12V power supply is connected directly to the motor drivers and to two parallel power-regulator circuits that step down to 5V. One of the 5V sources supplies power to a USB hub, and the other powers the BeagleBoard.

3. SOUND DESIGN
The central software for the audio control and haptics is Pure Data [7]. There are two primary audio patches involved in the instrument. One patch manages audio sample recording and playback and the other handles spatialization.

The spatialization patch uses a vector base amplitude panning (VBAP) object, which is included in the standard Pd-extended distribution [6]. The `vbap` object interprets azimuth, elevation, and spread angle parameters to generate appropriate gain multipliers for the signals being sent to each speaker. Each virtual sound mass is associated with an individual spatialization object so sounds may be panned independently.

Two instances of the audio recording/playback patch manage the sound samples associated with each virtual mass. When a button is pressed its previously stored sample crossfades with live audio from the 1/8" input jack. The audio is buffered and

NIME'11, 30 May–1 June 2011, Oslo, Norway.

loops when a user releases the record button. The pitch of a sample varies in proportion to the magnitude of the force exerted on its corresponding mass. This provides auditory feedback as a frequency wobble that indicates which of the two sound samples is being manipulated. This modulation can be "played" by gently wiggling a slider when a mass is attached.

4. HAPTIC MODELS

The haptic models for the Sound Flinger were developed in Pure Data (PD) and are based on Edgar Berdahl's haptic object library [3]. The `mass~` object is modified to modulo index the object's position to account for circular movement.

The slider handles are modeled with the `contact-detent~` object, which applies a force on the masses proportional to their distance from a handle. At a threshold distance this restoring force drops sharply to zero. The effect of this model feels as though a slider and mass are temporarily connected to a spring until the mass moves fast enough to break free.

It is possible to launch an attached mass by smoothly accelerating and then quickly stopping the slider. The mass will retain most of its momentum and break free of the detent region, continuing around the circle. It is possible to catch a mass by simply letting it pull a slider until its motion is sufficiently slowed by the sliders' friction.

5. RESULTS AND FUTURE WORK

We found the instrument to be very approachable, allowing novices with only a basic understanding of the device to experiment and immediately achieve interesting results. One common experience is gesture discovery, in which users develop repeatable sequences of interaction as they become more familiar with the instrument's behavior.

For example, if all sliders are held stationary while a mass is rotating around the sound field it will continue to circle indefinitely. Preventing the sliders from moving effectively eliminates all damping from the system. Another interesting gesture involves coaxing both masses onto a single slider, where the natural volatility of the mass-spring simulation causes the masses to oscillate in opposing directions. The modulation of the audio playback rates for each sound object gradually increases in magnitude as each mass gains momentum. Eventually a mass will break free and the system will return to a state of equilibrium.

In the future, we hope to extend this initial work and integrate feedback received from initial demonstrations. The most commonly requested feature is visual feedback to reveal the precise locations of the virtual masses, regardless of whether or not they are attached to a slider. This could be done with LEDs placed around the periphery that change color and/or brightness in relation to the positions of the virtual masses. A direct mapping from the gain multipliers generated by the spatialization patch to the brightness of the LEDs would provide appropriate visual feedback. Another potential improvement would be the addition of a separate headphone monitor connected directly to the instrument's audio input. This would allow users to more accurately cue sound samples from their input device of choice.

In addition to general design enhancements, we hope to observe people interacting and improvising with the instrument in a more public setting, such as a gallery or a concert. It would be useful to determine a timeline for gesture discovery based on trials with individuals of varying musical backgrounds. Furthermore, a systematic comparison between using the instrument with active versus inactive haptic feedback would provide insight into the significance of haptics for developing advanced performance techniques.

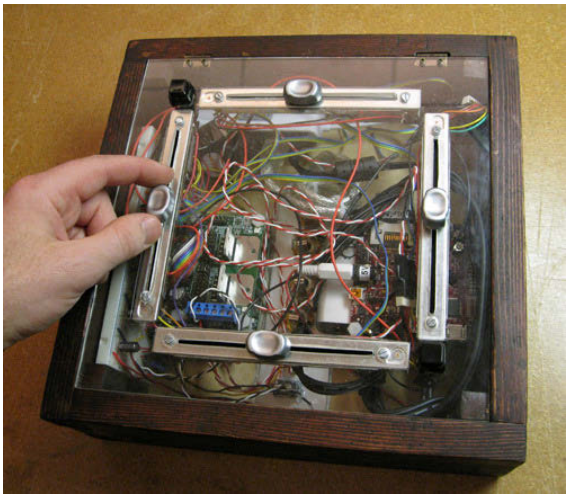


Figure 2. The Sound Flinger

6. ACKNOWLEDGMENTS

Special thanks to Ed Berdahl and Wendy Ju for their valuable guidance. Thank you to Bill Verplank, Max Mathews, Chris Chafe, Perry Cook, Matt Wright, and all of the additional people involved in developing the 250 course sequence that inspired this work.

Figure 3. Playing the Sound Flinger

7. REFERENCES

[1] Arduino Nano: http://www.arduino.cc/en/Main/ArduinoBoardNano

[2] Beagleboard Platform: http://beagleboard.org/

[3] Berdahl, E. Kontogeorgakopoulos, A. Overholt, D. HSP v2: Haptic Signal Processing with Extensions for Physical Modeling. *Proceedings of the Haptic Audio Interaction Design Conference*, Copenhagen, Denmark, September 16-17 2010, pp. 61–62.

[4] Planet CCRMA: http://ccrma.stanford.edu/planetccrma/software/

[5] Procyon Motor Driver Board: http://hubbard.engr.scu.edu/avr/boards/motordriverv10_manual.pdf

[6] Pulkki, V. Virtual Sound Source Positioning using Vector Base Amplitude Panning. *J. Audio Eng. Soc.* Vol 45, No. 6. June, 1997.

[7] Pure Data: http://puredata.info/

[8] Satellite CCRMA. http://ccrma.stanford.edu/~eberdahl/Satellite

[9] Verplank, B. Mathews, M. Sapp, C. A Course on Controllers. *Proceedings of the 2001 Conference on New Interfaces for Musical Expression.* 2001.

DAFT DATUM – AN INTERFACE FOR PRODUCING MUSIC THROUGH FOOT-BASED INTERACTION

RAVI KONDAPALLI
CCRMA
660 Lomita Dr
Stanford, California – 94305
ravik@ccrma.stanford.edu

BEN-ZHEN SUNG
CCRMA
660 Lomita Dr
Stanford, California – 94305
bsung88@stanford.edu

ABSTRACT
Daft Datum is an autonomous new media artefact that takes input from movement of the feet (i.e. tapping/stomping/stamping) on a wooden surface, underneath which is a sensor sheet. The sensors in the sheet are mapped to various sound samples and synthesized sounds. Attributes of the synthesized sound, such as pitch and octave, can be controlled using the Nintendo Wii Remote. It also facilitates switching between modes of sound and recording/playing back a segment of audio. The result is music generated by dancing on the device that is further modulated by a hand-held controller.

Keywords
Daft Datum, Wii, Dance Pad, Feet, Controller, Bluetooth, Musical Interface, Dance, Sensor Sheet

1. INTRODUCTION
In our preliminary research, we came across endeavors that linked movement to music [2][3]. Nintendo released a Power Pad in 1986 that also allowed users to play musical notes with their feet. It blended movement and music in a rudimentary way that did not allow for much musical variety. Dance Dance Revolution, the music video game, capitalizes on the aesthetic of dancing feet, but the music heard is not consequential to the movements of the user. DDR EAMIR [2] is a dance pad with prerecorded musical loops that can be triggered and synchronized in real-time by a user's feet. But long and prerecorded loops do not promote minute movements. In each case, there was either limited musical capability or restricted controllability and as a result, user expressivity was limited.

In designing Daft Datum, we sought to create a musical interface that translates highly expressive bodily gestures into equally expressive sound, effectively blending dance and music. Our design incorporates a commercially available 'dance pad' [11] and a 'Wii Remote' [6][7], which communicate with Pure Data (PD) for sound synthesis.

2. HARDWARE
Daft Datum includes a sensor sheet that has marked squares. When pressure is applied to these zones, the device sends out discrete information that can be interpreted by any software capable of reading HID information.

The Wii Remote, originally intended for use with the Wii Console, is a bluetooth device that pairs with many other bluetooth adapters. The remote is sturdy in its build and includes a set of sensors that transmit accurate sensor data. Examples include accelerometer data, button pressing/depressing and IR information.

The Beagle Board [5] is a low-power, low-cost single-board computer produced by Texas Instruments, with a 600MHz OMAP3530 processor, running Satellite CCRMA [1]. The Beagle Board runs PD and connects to the dance pad's sensor sheet and Wii Remote through USB and bluetooth respectively. The Beagle Board introduces autonomy and portability into the entire project, making it laptop/desktop machine-free.

3. PHYSICAL DESIGN
The idea was to prototype a device that would bear the weight of a dancing body. At the same time, the material covering the sensor sheet would have to be pliable enough to allow for accurate and low latency sensing of pressure from the user's feet. So, a body consisting of a wooden square enclosure was built, within which lay (in order from top to bottom) planks of thin wood, a sensor sheet and a sheet of dense foam.

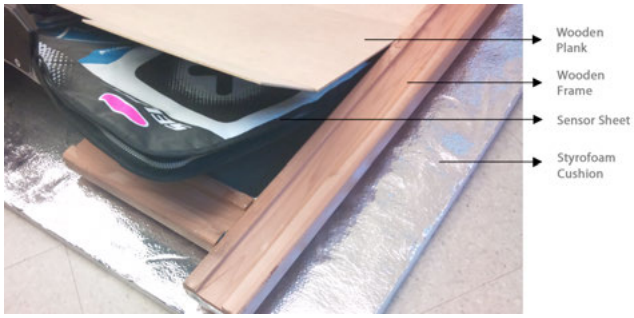


Fig. 1: Daft Datum

4. SOUND DESIGN
Given just eight sensors on the sheet, we asked ourselves how one could maximize the variety of sounds so that the user has enough range of expression for dancing out musically interesting pieces? Daft Datum incorporates multiple user modes, analogous to having a set of function keys on a computer. Each square on the pad is capable of producing two sounds, depending on the mode currently selected.

Fig. 2: Samples in Mode 1

The first mode features an assortment of recorded percussive samples (see Fig. 2) with which the user can create groovy, rhythmic base lines. In this mode, the percussion samples are assigned to the back row (from the perspective of the performer) in an arrangement that is suggestive of stomping out the beat. The two other rows feature non-percussive samples that supplement the percussive samples, adding variety to the base line. The second "synth" mode enables the user to compose melodies with notes from the D-minor scale. The sound synthesizer sums three sawtooth waves of different frequencies and post-processes them with a low-pass filter to produce a rich and crisp timbre.

5. THE WII REMOTE CONTROLLER

In order to further the sonic capabilities of the device, allow more expressivity of the dancer and enrich the interaction between the two, the notion of a handheld controller was conceived with the intention that the hands would shape the music produced by the feet. To this end, the Wii Remote was particularly appropriate, as it not only satisfied all these requirements but also communicated flawlessly with PD.

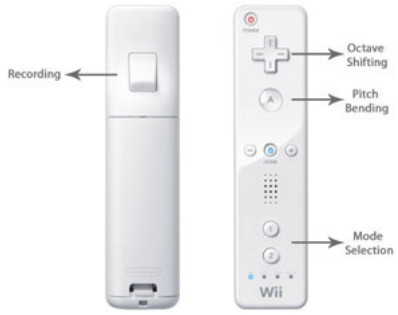


Fig. 3: Functions of the Wii Remote buttons

The buttons on the Wii Remote were intuitively mapped to switch modes, bend pitches, shift octaves and record segments of music that could then be looped (see Fig. 3).

1) The buttons 1 and 2 switch the sensor sheet mappings between the corresponding modes.
2) Pitch bending, activated when the 'A' button of the remote is held-down, is the product of accelerometer data that changes the frequency of a given note as it is being played.
3) The octave register of notes currently played by the feet are shifted by pressing the "up/down" buttons – multiplying the note's frequency with a factor of 2 or 0.5, respectively.
4) Pressing the "B" button of the controller activates the recorder functionality in the patch. Holding the button down records any sounds currently being played; releasing the button stops the recording and starts looping the segment that has just been recorded.

6. COMMUNICATION BETWEEN DEVICES

The sensor sheet is plugged to a USB port on the Beagle Board and is recognised in Pure Data using the `hid` [7] object. The Wii Remote is paired with a bluetooth dongle plugged into another USB port on the Beagle Board. The `wiimote` [4][6] object is used to read data from the dongle. The readings from both USB ports were simultaneously processed in a custom patch. This patch sends out audio through the regular system-out.

7. CONCLUSION AND FUTURE WORK

Although the current physical design of Daft Datum suffices for prototyping purposes, we envision a future version of this interface that is sturdier and portable. Ideally, instead of carrying and setting up four separate layers of material, there would be a single platform that would also enclose the Beagle Board.

Another important development that we envision Daft Datum to have is a self-contained sound synthesis module, in which all sounds being produced are synthesized using Pure Data objects and not pre-recorded samples, specifically for the Mode 1, as we hear them now.

In conlusion, Daft Datum seeks to provide a medium for musical composition via dance, an art form which is cross-cultural [6]. It was aimed at a broad range of audience and does not require previous musical experience. However, with some practice, there is room for complexity in sound design and the user can create more intricate musical statements.

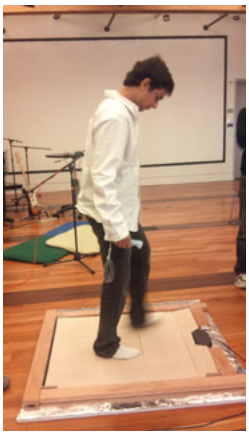


Fig 4. Performance on Daft Datum

8. ACKNOWLEDGEMENTS

We would like to thank our instructors Edgar Berdahl and Wendy Ju for the technical and design advice.

9. REFERENCES

[1] Berdahl, Ed & Wendy Ju – Satellite CCRMA
http://ccrma.stanford.edu/~eberdahl/Satellite
[2] DDR EAMIR
http://www.vjmanzo.com/clients/eamir/ddr.htm
[3] http://www.youtube.com/watch?v=F8NC2EZ8cCk
[4] Homepage for the CWiid package
http://abstrakraft.org/cwiid/
[5] Product Details – Beagle Board
http://beagleboard.org/hardware
[6] Smith, Jacob. I Can See Tomorrow In Your Dance: A Study of Dance Dance Revolution and Music Video Game
[7] Steiner, Hans C., HID for PD http://at.or.at/hans/pd/hid.html
[8] Wii Remote from Wikipedia
http://en.wikipedia.org/wiki/Wii_Remote
[9] Wii Controllers
http://www.nintendo.com/wii/console/controllers
[10] Wozniewski, Mike – Wiimote Package for PD
http://mikewoz.com/pd-stuff.php
[11] USB DDR Dance Pad
http://dealextreme.com/feedbacks/BrowseReviews.dx/sku.4234

10. APPENDICES

1. Videos of performances on Daft Datum
   http://www.youtube.com/watch?v=zqpSiAxKDMI
   http://www.youtube.com/watch?v=xhld60K7d2w

Strike on Stage: a percussion and media performance

Charles Martin
Department of Music and Media,
Luleå Technical University
Piteå, Sweden
cpm@charlesmartin.com.au

Chi-Hsia Lai
Media Lab, Department of Media,
School of Art and Design, Aalto University
Helsinki, Finland
me@laichihsia.com

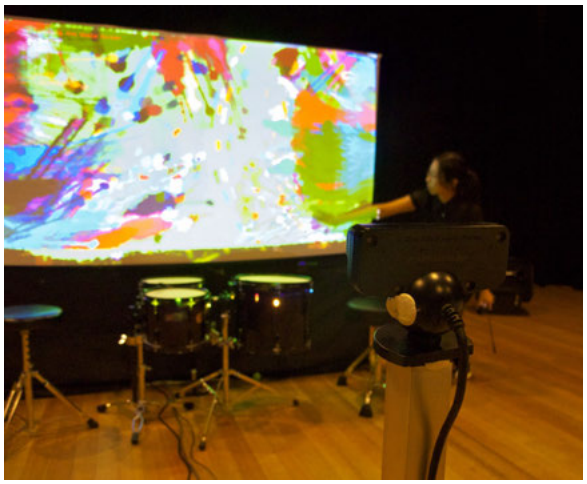


Figure 1: *Strike on Stage* in performance.

ABSTRACT

This paper describes *Strike on Stage*, an interface and responding audio-visual performance work developed performed in 2010 by percussionists and media artists Hsia Lai and Charles Martin. The concept of *Strik Stage* is to integrate computer visuals and sound int improvised percussion performance. A large projectior face is positioned directly behind the performers, wh computer vision system tracks their movements. The s allows computer visualisation and sonification to be dir responsive and unified with the performers' gestures.

Keywords

percussion, media performance, computer vision

1. INTEGRATING MEDIA PERFORMA

Strike on Stage is an interface and corresponding a visual performance developed and performed in 201 percussionists and media artists Chi-Hsia Lai and Ch Martin. The aim of the work was to integrate comput suals and sound into an improvised percussion performance. This integration takes place in two ways. First, the performers' gestures are linked to computer audio and visuals through a computer vision system and microphone. Secondly, the presentation of the performance is unified. The projection screen and loudspeakers are placed immediately behind the performers and the computer visuals and audio is designed in such a way that, to the audience, they appear to be a natural augmentation of the performers' forms, instruments and gestures.

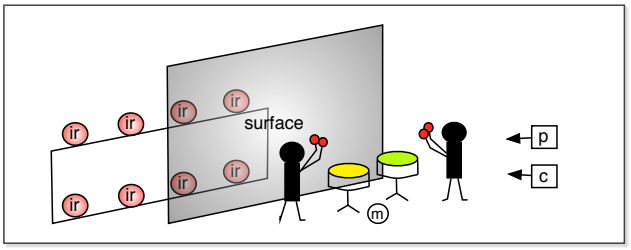


Figure 2: Setup diagram for *Strike on Stage*.

The work is a successor to Lai's performance work *Hands on Stage* [2], developed at the Australian National University. In *Hands on Stage*, Lai used a computer vision surface similar to that commonly used for *reacTIVision* [1]. In the case of *Hands on Stage*, the focus was not on making a touchable GUI, since there was no projector under the acrylic surface, but rather an extended musical and media instrument. Sounds were produced by the shadow of the player's hands on the surface, detected by a camera, and contact microphones amplified the sound of the player scratching and tapping the surface. *Hands on Stage* also had a visual component influenced by the image of the player's hands which was projected onto an external screen.

Strike on Stage was conceived to further the artistic direction of *Hands on Stage* while addressing some of the limitations of the interface. Whereas *Hands on Stage* was designed for a solo performer using only their hands, *Strike on Stage* was designed for two performers, using their whole arms, bodies and drum sticks or other percussion implements to control the performance. The video projection was to be integrated into the performance surface so that the audience's focus is not divided between the performers and an external screen.

2. THE INTERFACE

The setup for *Strike on Stage* is centred around a floor-standing projection screen made from thin fabric (denoted 'surface' in figure 2). An array of 8 infra-red LED security lights (denoted 'ir') is placed behind the surface and the lights aimed to provide an even illumination of the screen.

The performers and instruments are positioned directly in front of the screen. Infra-red light passes through the screen from behind enabling an infrared sensitive camera

Figure 3: One interactive scene in the performance.

('c') placed around 3 meters in front of the performers to see a clean silhouette of the performers and instruments. A projector ('p'), outputting mainly visible light is also placed in front of the screen to project onto both the screen and performers. Additionally, a microphone ('m') is placed underneath the instruments and loudspeakers are hidden with all computer equipment behind the screen and IR lights.

This setup gives a clean image for computer vision purposes and integrates the projection surface with the performers and instruments. Furthermore, the screen and lighting array is lightweight and portable and can be reproduced relatively cheaply for future projects.

Since the computer equipment and loudspeakers are hidden behind the screen, the setup for *Strike on Stage* presents a relatively minimalistic stage presence to the audience.

3. THE PERFORMANCE

The performance component of *Strike on Stage* consists of a series of interactive set pieces created in openFrameworks [3] and SuperCollider [4], along with percussion improvisations. The work focuses on exploring how the interactive environment can augment the percussive gesture of 'striking' an instrument.

Percussion performance is often characterised by the visual drama of striking instruments as much as their sound. As a result, part of a percussionist's individual style is their approach to striking instruments, both in order to produce particular sounds and to emphasise elements of the performance to the audience.

The interactions developed for *Strike on Stage* augment and react to these 'percussive gestures'. The technical approach for detecting these movements from blob-tracking algorithms provided in openFrameworks is simple. The performers are positioned side-on to the screen and facing each other (as in figure 2). This means that the tip of the left performer's sticks or hands are the rightmost point of their silhouette and similarly for the leftmost point of the right performer's silhouette. Percussive gestures can be detected by tracking the acceleration of these points. When a drumstick bounces off a drum it has a high acceleration away from the drum at the point of impact.

In one of the most effective interactions in *Strike on Stage*, manipulated cymbal and gong samples were played in SuperCollider each time a sharp acceleration was detected at the tips of each performer's sticks. Lines were projected around the edges of the performer's silhouettes that varied their length with the acceleration of that point on the silhouette (shown in figure 3). The result was that the performers could not only play computer sounds by striking their physical instruments but also by 'air drumming'. Since the rationale for triggering extra sounds was simple the performers could control the extent of the effect and play with the interaction in their improvisation. From the audience's perspective, the performers appeared to be surrounded by constantly shifting spines that shot out in synch with the energy of their motions.

Other interactions in the work used the performer's motions to trigger and manipulate field recordings and photographs, both taken by the performers while producing the work. The result was a collage linking the collaborative improvisation with audio and visual textures from the development of the work.

The overall tone of the work was a playful exploration of the affordances of the screen and computer sound. Artistically the work emphasised the 'strike' movement of playing percussion instruments and made connections between the performer's movements on stage and their lived experience as creators of the work.

4. CONCLUSIONS AND FURTHER WORK

Strike on Stage was performed in 2010 as *Strike on Stage 1.0*, and performances were held at Belconnen Arts Centre, Canberra, NIME2010, Sydney and the Australasian Computer Music Conference 2010, Canberra. The work was also converted into a 'micro' version with a much smaller screen and only one IR light. These performances are documented on the project's blog[1].

These performances proved that the setup for *Strike on Stage* was viable in a range of performance conditions even when setup time was extremely limited. Furthermore, feedback from the audience confirmed that the performance method was interesting and effective.

The strategy for capturing 'percussive gestures' from blob-tracking algorithms was reasonably effective, but there is much scope to explore other connections between the performers' movements and computer sound and visuals.

Although there are plans to revise *Strike on Stage* with a new version in 2011 the same techniques could inspire other artistic projects. We imagine a collaboration with a composer or an ensemble with multiple 'micro' screens and small projectors.

5. ACKNOWLEDGMENTS

This project was supported by the A.C.T. Government, Australia.

6. REFERENCES

[1] M. Kaltenbrunner and R. Bencina. reactivision: a computer-vision framework for table-based tangible interaction. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 69–74, New York, USA, 2007. ACM.

[2] C.-H. Lai. Hands on stage: A sound and image performance interface. In *Proceedings of the 9th Conference on New Interfaces for Musical Expression*, June 2009.

[3] The openframeworks homepage - "an open source c++ toolkit for creative coding". Available from http://www.openframeworks.cc/ [cited January 2011].

[4] The supercollider homepage - a real time audio synthesis programming language. Available from http://www.audiosynth.com/ [cited January 2011].

[1] `http://strikeonstage.posterous.com`

Gestural Embodiment of Environmental Sounds: an Experimental Study

B. Caramiaux, P. Susini, T. Bianco, F. Bevilacqua, O. Houix, N. Schnell, N. Misdariis

IMTR and PDS Team
Ircam - CNRS
1 place Igor Stravinsky
75004, Paris, France

baptiste.caramiaux@ircam.fr

ABSTRACT

In this paper we present an experimental study concerning gestural embodiment of environmental sounds in a listening context. The presented work is part of a project aiming at modeling movement-sound relationships, with the end goal of proposing novel approaches for designing musical instruments and sounding objects. The experiment is based on sound stimuli corresponding to "causal" and "non-causal" sounds. It is divided into a performance phase and an interview. The experiment is designed to investigate possible correlation between the perception of the "causality" of environmental sounds and different gesture strategies for the sound embodiment. In analogy with the perception of the sounds' causality, we propose to distinguish gestures that "mimic" a sound's cause and gestures that "trace" a sound's morphology following temporal sound characteristics. Results from the interviews show that, first, our causal sounds database lead to consistent descriptions of the action at the origin of the sound and participants mimic this action. Second, non-causal sounds lead to inconsistent metaphoric descriptions of the sound and participants make gestures following sound "contours". Quantitatively, the results show that gesture variability is higher for causal sounds that non-causal sounds.

Keywords

Embodiment, Environmental Sound Perception, Listening, Gesture Sound Interaction

1. INTRODUCTION

In the context of music playing as well as music listening, movements and actions related to musical stimuli can be seen as the embodied manifestation of sound/music perception and cognition [14, 7]. In the cognitive neuroscience literature, previous works have shown some evidences for music embodiment in the auditory-motor systems interaction during music performance (see [17] for a review). For instance, people naturally tap the beat while listening to a piece of music and often anticipate the rhythmic accents [11, 12]. In [4], the authors investigate a more abstract relationship between body motion and music that is examining whether changes in musical parameters evoke corresponding changes in listeners' spatial and kinetic imagery. In parallel, a need for a coherent typology of music-related gestures or actions has emerged [2]. A movement reacting to sonorous stimuli can be qualified as sound-accompanying gestures [10], i.e. gestures that are not involved in the physical production of sound but rather are reflecting some important aspects in sounds.

Godøy et al. have conducted two experimental studies showing two sub-categories of sound-accompanying gestures: gestures that mimic instrumental performances [9] and sound-tracing gestures [8]. While the first study is concerned by musical piece stimuli, the second involves a larger set of sounds from musical instruments, electronic sounds or environmental sounds (taken as *concrete* sounds in the sense of Schaeffer [15]).

Through their explorative works, Godøy et al. have highlighted two interesting strategies in music embodiment: *mimicking* and *tracing*. However, they were studied independently with two distinct experimental protocols. We believe that both strategies constitute an important dichotomy in gestural sound embodiment and precisely when considering environmental sounds. To that extent, it seems pertinent to considered them jointly. The experiment presented in this paper aims to characterize both *mimicking* and *tracing* strategies through an unique experimental protocol.

Computational characterization of these two strategies related to environmental sounds can be insightful for the design of virtual instruments and sound design tools. Mimicking can be transcribed as the excitation of a specific physical model while tracing can be transcribed as the instantaneous mapping between gesture features and audio features. Both strategies can lead to a wide range of applications for sonic interaction design as well as future theoretic studies.

The paper is organized as follows. The next section aims at placing our contribution in the state of the art. As far as we know, very few works exist on characterization of embodied listening of environmental sounds. Therefore, the related work is focused on sound perception and listening strategies. Our methodology is reported in section 3. This is the starting point for our experimental study that is divided into two steps. First we present the sound stimuli in section 4 then we define an experimental protocol to evaluate our hypothesis in section 5. Results are presented inside of each section. Finally we conclude in section 6 giving some ongoing short-term perspectives.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. RELATED WORKS

In environmental sound perception, Gaver in [5, 6] has proposed the distinction between *musical listening* and *everyday listening*. In musical listening the listener focuses on acoustic qualities and other musical aspects of sound while in everyday listening the listener focuses on causal aspects. In the task of categorization of sounds, this suggests that some people will consider as similar two sounds with the same acoustic characteristics and others will consider as similar two sounds with the same cause. Following these previous studies, Lemaitre et al. in [13] have shown the categorization of environmental sounds is influenced by the listener's expertise and the sound identification (i.e. if the cause that has produced the sound is identifiable or not). They showed that in categorization task, people will more frequently base their choice on acoustic characteristics if the identification of the cause is difficult (i.e. the causal uncertainty is high). On the other hand, people will frequently use the sound's cause as categorization criterion if the causal uncertainty is low.

Our contribution is to propose an experiment that analyses how people embody musical or everyday listening of environmental sounds. The methodology is exposed in the next section.

3. HYPOTHESIS

Previous works [9, 8, 3] roughly depict two categories for gesture embodiment of environmental sounds: gestures mimicking the action that has produced the sound and gestures following (or tracing) the temporal evolution of the perceived sound features. In the following we will use the terms *symbolic* referring to the gestures from the first category and *morphologic* referring to the gestures from the second category. This terminology emphasizes the distinction between the symbol and the shape. These two terms are not established and a deeper discussion about their use is part of our prospective works.

Consider the following experimental methodology. We propose to consider causal sounds and to synthetically take off the causality by an audio process (that roughly corresponds to retain the global energy evolution whereas timbre characteristics are flatten). Then we ask for people to associate gestures while listening these causal and non-causal sounds.

The goal is to analyze the gesture and sound data to explore the following hypothesis: *causal* sounds imply *symbolic* gestures and *non-causal* sounds induce *morphologic* gestures?

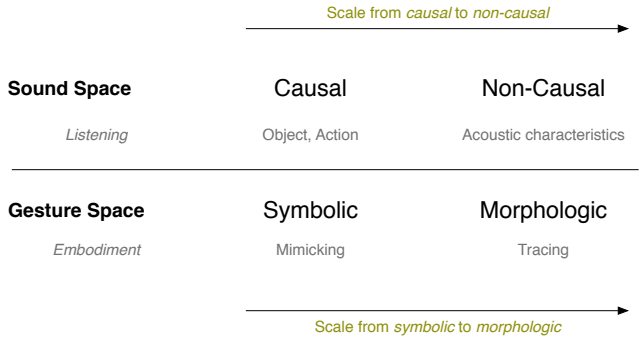


Figure 1: Scheme of the global working context of the experimental study

4. SOUND SELECTION PROCEDURE

The sounds used in the study belong to a domestic context (usual objects found in a kitchen), to ensure that the sources of the sounds were likely to be known to all listeners [13]. Each sound identification is calculated through the causal uncertainty index (noted H_{cu}) [1, 13] that measures the identification of the cause in terms of action and/or object verbalized description. Each sound has a H_{cu} index scaled between 0 (i.e. all the participants provided the same description of the sound in terms of action or object) and 4.75 (all the participants provided a different description in terms of action or object). However, the procedure of measuring H_{cu} is very time-consuming and needs for a precise semantic analysis of verbalizations. Instead the authors propose to measure the confidence in the identification by an usual scale between 1 and 5.

1. "I don't know at all"
2. "I am really not sure"
3. "I hesitate between several causes"
4. "I am almost sure"
5. "I perfectly identify the cause of the sound"

Lemaitre et al show that the resulting measure is correlated to H_{cu} even if both measures do not provide exactly the same information. From this previous study, we have selected the ten most identified sounds (low H_{cu}) in the kitchen sounds database in order to define a first corpus, namely the "causal" sounds. Having the corpus of causal sounds, we build a second corpus by applying an audio process transforming the sounds taken from the first corpus. We design a sound transformation that takes the original causal sound and returns a sound with the same energy evolution but having occulted some of the timbre aspects. The transformation is convolution-based and is illustrated by figure 2. In this figure, the reader can see that the temporal evolution of the mel cesptrum remained whereas timbre characteristics of original sounds are flatten.

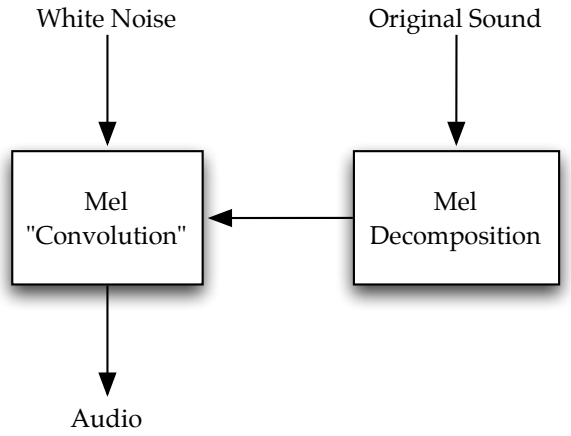


Figure 2: Scheme of the sound transformation used in the pre-experiment. Original sound is analyzed according Mel decomposition. Then, mel coefficient evolutions are used to convolve with white noise. It results an audio stream with the same energy distribution as the original sound but without timbre.

The experiment accounted for 21 non-expert candidates that have rated on a scale from 1 to 5 their confidence in identifying the action that have produced the sounds. Eleven candidates were assigned to the non-transformed sound corpus and the other ten to the transformed one.

The results can be seen on figure 3. The figure shows the statistics for each sound. Plot on the left corresponds to the original sounds. Plot on the right corresponds to the transformed sounds. Black solid horizontal lines are the median rate and boxes are illustrating the deviation between the first and the third quartile. Black dashed lines are reporting min and max values.

The results show that for some sounds of the transformed corpus the candidates are still confident in their identification of the sound cause (e.g. sounds 1 and 2). However, other sounds are efficiently non-identifiable (e.g. sounds 8 and 9). We select four sounds that represent the best the effect of the audio transformation and having different temporal profiles. The resulting corpus contains 8 sounds corresponding to: NT 4, NT 6, NT 8, NT 9 and T 4, T 6, T 8, T 9 (where NT=non-transformed and T=transformed) corresponding to:

(4) glass impact
(6) pouring rice
(8) screwing a bottle cap
(9) squeezing a can

Median rates for the selected sounds are: 5 (NT4), 2.5 (T4); 5 (NT6), 3 (T6); 2.5 (NT8), 1.5 (T8); 4.5 (NT9), 1 (T9).

5. EXPERIMENTAL PROTOCOL

In this section we present the experimental protocol. Since we have two corpuses, the same protocol is used for each corpus and each candidate participates to the experiment for only one of the two corpuses.

5.1 Method

5.1.1 Task

The task is presented as follows. "*You must perform a gesture associated to the sound you will listen to. Here "associated" means performing gestures that mimic the action producing the sound or that follow temporal evolution of the sound*". Two fixed examples for the different strategies that can be adopted in the performance are illustrated by the examiners. The strategies are explicitly told to the participants to avoid participants to be lost when being faced to such a non-usual experience. The experiment continues with two phases: the performance and the interview.

5.1.2 Phase 1: Performance

Only one of the two corpuses is used per candidate. The participants are asked to perform gestures synchronously to the sound they are listening. For each sound of the corpus, there are three sequential steps: *training*, *selecting*, *validating*. In the first step, the participant can listen to the sound any number of times. Synchronously, any number of rehearsals can be performed in order to find the gesture that is, for the participant, well associated to the sound. When the candidates feel confident, they select the associated gesture (so-called *candidate gesture*). The final step is the validation of the candidate gesture. The participant must perform three times exactly the same gesture. This step validates that the candidate gesture is stabilized. The whole performance phase is recorded by a video camera.

5.1.3 Phase 2: Interview

The interview is an *auto-confrontation* of the participants with their performance [16]. Together with the participants we sequentially visualize the videos corresponding to each sound. Only the candidate gestures are watched (i.e. four videos). For each candidate gesture we ask questions that allow the participants to verbalize their action. First we discuss what came spontaneously to their mind when they first listened to the sound. Then we discuss the gesture they performed (e.g. *was it difficult to find the gesture? what are the different steps in your gesture?* etc.). Finally, we discuss the relationships between the performed gesture and the listened sound (e.g. *did you try to be synchronous?* etc.).

The aim is to help the analysis of the data collected during the experiment. Verbalization given by the participants informs us on their intentions during the performance: for instance if they tried to mimic a specific action or to follow acoustic features; how they can describe the listened sound; if they were comfortable with the interface etc.

5.1.4 Data collection

Participants. Twenty-two non-musician subjects participated to the experience, which took place at Ircam between August and October 2010. In a mixed between-within design, two groups, of 11 subjects each, performed either on the Non-Transformed, or on the Transformed sound corpus stimuli. The experiment took approximately one hour, and the participation was retributed with a nominal fee.

Material. The hand's position was captured by tracking on-hand placed markers with an ARTtrack motion capture system at 100Hz sample rate. No other motion capture interface was used during the experiment. The sound stimuli were monophonic and had 16-bit resolution and a sampling rate of 44.1kHz. A video camera recorded each performance. Motion, audio and video were recorded synchronously at each trial using the real time programming environment Max/MSP.

5.2 Results

5.2.1 Interviews: mimicking and tracing

First, we examine the interviews for participants having listened to the non-transformed corpus. Globally, the participants do not succeed to describe the sounds' characteristics but rather describe the action that has produced the sound. Sound descriptions show that gestures associated to the sounds focus on the action in interaction with an object. While they do not accurately describe the object, they are more consistent on the actions. The terminology used to describe each sound can be synthesized as: sound 1, *to hit* (70%); sound 2, *to pour* (85%); sound 3, no clear terminology *to pull*, *to scrap*, *to push*; sound 4, *to squash* (85%). The gesture associated to the sounds corresponds to the action described. Finally, all participants have imagined manipulating an object while they were performing their gesture.

Second we examine the interviews for participants having listened to the transformed corpus. It appears that the participants have not precisely recognized an action or an object. The cognitive representation associated to the sounds is often metaphorical and with large variations across the candidates. Gestures associated to the sounds are described as representations of the corresponding metaphors. The time evolution of the sound characteristics are often referred in the descriptions. To conclude, the interviews reveal that the metaphor associated to a sound emanates from the sound characteristics.

5.2.2 Performed gesture characterization

We are interested in analyzing the gesture variability for each sound from each corpus: *non-transformed* and *transformed*. We choose in a first step to take into account the velocity, found in a previous study as one of the important gesture parameter. Temporal evolution of sound and sensation of energy in our body are linked by the gestural representation of sound during the experiment. Considering

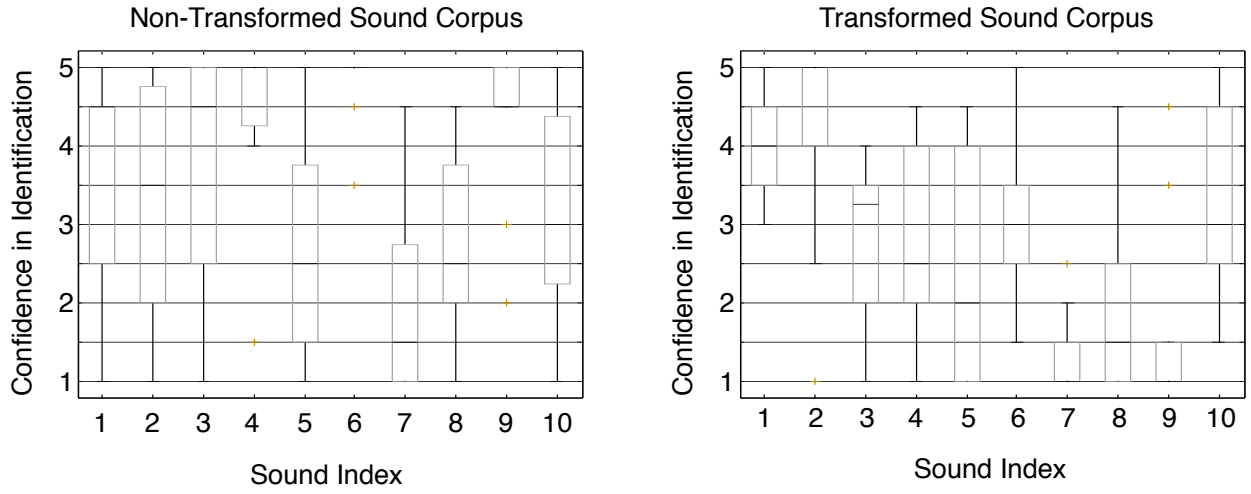


Figure 3: Building corpus. Rates are given for each sound (from sound 1 to sound 10 either for causal sounds or non-causal sounds). Plots depict statistics on resulting rates. Crosses are outliers. Black solid horizontal lines are the median rate and gray boxes are illustrating density between first quartile and the third quartile. Black dashed lines are reporting min and max values.

velocity allow us to be position and direction independent as well as focusing on kinetic energy. A further detailed analysis that consider other gesture parameters (like accelerations or jerks) is left as future work.

Figure 4 illustrates all the performances for each sound from each corpus. Each plot represents from top to bottom: The waveform for the non-transformed sound i; The *candidate* gestures associated to the non-transformed sound i by all the participants: upper bound is the third quartile limit, lower bound is the first quartile limit and the curve is the median evolution; The corresponding transformed sound i; The *candidate* gestures associated to the transformed sound i by all the participants. Gesture variability is computed as the mean and variance of the density range defined as the upper bound minus the lower bound. Results are given in table 1.

	Sound #1	Sound #2	Sound #3	Sound #4
NT	0.920 ±0.285	1.235 ±0.134	0.778 ±0.146	1.144 ±0.256
T	0.591 ±0.180	0.829 ±0.095	0.863 ±0.423	0.605 ±0.111
$\frac{NT-T}{NT}$ (%)	*-35.8*	*-32.9*	*+11.0*	*-47.2*

Table 1: Global cumulative variance

One can see that the gestures performed while listening to the transformed sounds 1, 2 and 4 are less varying than the ones associated to non-transformed sounds. The means are significantly distinct (according to a t-test with α level set to .01). However, there is no significant difference in variability between gestures associated to non-transformed and transformed sound 3 (that is *screwing a bottle cap*). Actually, sound 3 (referring to sound 8 in figure 3) was the less contrasted from the set of selected transformed sounds: the median of confidence rate for non-transformed sound 3 was 2.5 while the median of confidence rate for its transformed version was 1.5. A greater gesture variability for causal sounds than non-causal sounds could be interpreted as follows. When participants identify the sound as its cause, each participant has their own manner to represent the cause. Otherwise, when participants identify the sound by its acoustic characteristics, each participant has a common reference to gesturally represent the sound.

6. CONCLUSION

The aim of this study was to better understand the dichotomy that can exist in gestural environmental sound embodiment. We establish a methodology based on two environmental sound corpuses (non-causal and causal sounds) used as stimuli for candidates. They had to associate a gesture for each sound from one corpus and verbalized their action during an interview. Results show that verbal description of the causal sounds are consistent and they comment their gestures as mimicking the cause whereas verbalization for non-causal sounds do not show a particular consensus in the sound identification. Interestingly, quantitative analysis on gesture data shows that gesture variability is lower for non-causal sounds than for causal sounds. A first interpretation is that people are consistent in the identification of action but the gestural representation of action is highly subjective (because some of these actions are commonly used in the everyday life). On the contrary, when the mental image of the sound cause is confused, the reference becomes the sound itself that is common to all the participants.

Prospective works will go further in the analysis of the terminology and gesture analysis as well as comparing gesture data to sound data. Another short-term perspective is the analysis of a second phase (not described in this paper) that consists in gestures performed on concatenation of the sounds taken from the two corpuses of causal and non-causal sounds presented in this paper.

7. ACKNOWLEDGMENTS

We acknowledge partial support from the project Interlude -ANR -08-CORD-010 (French National Research Agency).

8. REFERENCES

[1] J. Ballas. Common factors in the identification of an assortment of brief everyday sounds. *Journal of Experimental Psychology: Human Perception and Performance*, 19(2):250–267, 1993.

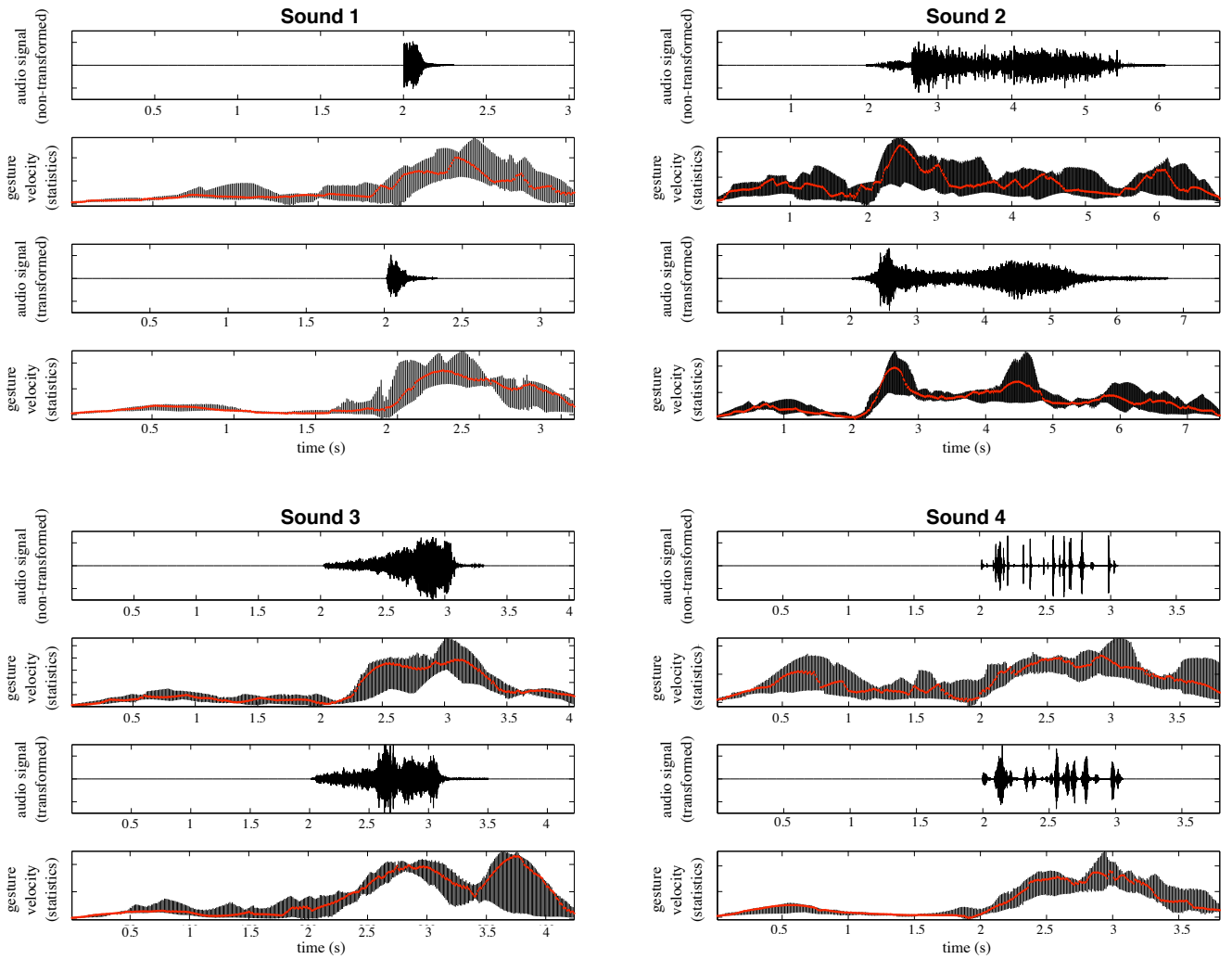


Figure 4: Gestures' velocity associated to each sound from each corpus. Each plot represents from top to bottom: The waveform for the non-transformed sound i; The *candidate* gestures associated to the non-transformed sound i by all the participants: upper bound is the third quartile limit, lower bound is the first quartile limit and the curve is the median evolution; The corresponding transformed sound i; The *candidate* gestures associated to the transformed sound i by all the participants.

[2] C. Cadoz and M. M. Wandereley. Gesture-music. *Trends in Gestural Control of Music*, 2000.

[3] B. Caramiaux, F. Bevilacqua, and N. Schnell. Mimicking sound with gesture as interaction paradigm. Technical report, IRCAM - Centre Pompidou, 2010.

[4] Z. Eitan and R. Granot. How music moves: Musical parameters and listeners' images of motion. *Music perception*, 23(3):221–248, 2006.

[5] W. Gaver. How do we hear in the world? explorations in ecological acoustics. *Ecological psychology*, 5(4):285–313, 1993.

[6] W. Gaver. What in the world do we hear?: An ecological approach to auditory event perception. *Ecological psychology*, 5(1):1–29, 1993.

[7] R. Godoy, A. Jensenius, and K. Nymoen. Chunking in music by coarticulation. *Acta Acustica united with Acustica*, 96(4):690–700, 2010.

[8] R. I. Godøy, E. Haga, and A. R. Jensenius. Exploring music-related gestures by sound-tracing: A preliminary study. In *Proceedings of the COST287-ConGAS 2nd International Symposium on Gesture Interfaces for Multimedia Systems (GIMS2006)*, 2006.

[9] R. I. Godøy, E. Haga, and A. R. Jensenius. Playing "air instruments": Mimicry of sound-producing gestures by novices and experts. In *Lecture Notes in Computer Science*. Springer-Verlag, 2006.

[10] A. R. Jensenius, M. Wanderley, R. I. Godøy, and M. Leman. Musical gestures: concepts and methods in research. In *Musical gestures: Sound, Movement, and Meaning*. Rolf Inge Godoy and Marc Leman eds., 2009.

[11] E. Large. On synchronizing movements to music. *Human Movement Science*, 19(4):527–566, 2000.

[12] E. Large and C. Palmer. Perceiving temporal regularity in music. *Cognitive Science*, 26(1):1–37, 2002.

[13] G. Lemaitre, O. Houix, N. Misdariis, and P. Susini. Listener expertise and sound identification influence the categorization of environmental sounds. *Journal of Experimental Psychology: Applied*, 16(1):16–32, 2010.

[14] M. Leman. *Embodied Music Cognition and Mediation Technology*. Massachusetts Institute of Technology Press, Cambridge, USA, 2008.

[15] P. Schaeffer. *Traité des Objets Musicaux*. Éditions du Seuil, 1966.

[16] J. Tardieu. *De l'ambiance à l'information sonore dans un espace public*. PhD thesis, Université Pierre et Marie Curie, 2006.

[17] R. Zatorre, J. Chen, and V. Penhune. When the brain plays music: auditory–motor interactions in music perception and production. *Nature Reviews Neuroscience*, 8(7):547–558, 2007.

Listening to Your Brain: Implicit Interaction in Collaborative Music Performances

Sebastián Mealla
Music Technology Group
Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona, Spain
sebastian.mealla@upf.edu

Aleksander Väljamäe*
Laboratory of Brain-Computer Interfaces
Graz University of Technology
Krenngasse 37
8010 Graz, Austria
aleksander.valjamae@tugraz.at

Mathieu Bosi
Music Technology Group
Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona, Spain
mbosi@gmail.com

Sergi Jordà
Music Technology Group
Universitat Pompeu Fabra
Roc Boronat, 138
08018 Barcelona, Spain
sergi.jorda@upf.edu

ABSTRACT

The use of physiological signals in Human Computer Interaction (HCI) is becoming popular and widespread, mostly due to sensors miniaturization and advances in real-time processing. However, most of the studies that use physiology-based interaction focus on single-user paradigms, and its usage in collaborative scenarios is still in its beginning. In this paper we explore how interactive sonification of brain and heart signals, and its representation through physical objects (*physiopucks*) in a tabletop interface may enhance motivational and controlling aspects of music collaboration.

A multimodal system is presented, based on an electrophysiology sensor system and the Reactable, a musical tabletop interface. Performance and motivation variables were assessed in an experiment involving a test "Physio" group (N=22) and a control "Placebo" group (N=10). Pairs of participants used two methods for sound creation: implicit interaction through physiological signals, and explicit interaction by means of gestural manipulation. The results showed that pairs in the Physio Group declared less difficulty, higher confidence and more symmetric control than the Placebo Group, where no real-time sonification was provided as subjects were using pre-recorded physiological signal being unaware of it. These results support the feasibility of introducing physiology-based interaction in multimodal interfaces for collaborative music generation.

Keywords

Music, Tabletops, Physiopucks, Physiological Computing, BCI, HCI, Collaboration, CSCW, Multimodal Interfaces.

*Also at SPECS Laboratory, Universitat Pompeu Fabra. Roc Boronat 138, 08018 Barcelona, Spain.

1. INTRODUCTION

In recent years, physiology-based systems have led to implicit models of interaction where the user's physiological signals, such as brain waves, electro-dermal activity (EDA) or heart rate, are monitored, mapped and transformed in commands to control devices and applications [23]. This interaction paradigm is based on internal states of the human body and has been explored by different disciplines such as cognitive psychology, neuroscience, physiological computing, enactive media and HCI. However, most of these studies are focused on single-user scenarios, either in clinical rehabilitation [27], or in communication and control applications [4]. At the same time, the use of electro-physiological systems in collaborative scenarios and Computer-Supported Collaborative Work (CSCW) is still scarce.

In this paper we present a collaborative music system that combines implicit interaction based on physiology sensing and explicit interaction based on a tangible interface for real-time sound generation and control (Reactable) [13]. This multimodal system displays physiological signals through sound, graphics and physical objects (*physiopucks*) which can be manipulated by physiology emitters and their partners. We hypothesize that such use of physiological signals via (*physiopucks*) will enhance motivational and controlling aspects of music creation in collaborative scenarios.

The study of HCI systems based on the combination of physiological signals and tabletops has not been widely explored. We are only aware of two similar studies using physiological signals and tabletops [28] [9], which nonetheless lack the collaborative and musical aspects that this paper aims to analyze in order to contribute to the understanding of such a paradigm.

To assess the effect of physiology-based interaction in music collaboration using the aforementioned system, task-oriented experiments between pairs of participants were carried on. Performance and motivational aspects of music collaboration were assessed using self-report methods.

2. STATE OF THE ART

2.1 Physiology-based Music

In the process of designing a physiology-based interface, specific body states are mapped to an explicit display technique [1]. For example, this can be achieved through interactive sonification, which allows the exploration of physiological

signals by their adaptive transformation into sound [10].

Research on sound and music computing pioneered the use of bioelectrical signals in interactive systems. Rosenboom's implementations of physiological measures for music generation are among the first outstanding works in the field. His musical systems presented parameters and textures driven by electroencephalography (EEG) and heart rate, among other physiological techniques [25].

More recent research associates EEG-acquired data with musical imagination [20], leading to new techniques and devices, such as Miranda's Brain-Computer Music Interface (BCMI) *Piano System* that trains the computer to identify EEG patterns associated with cognitive musical tasks, or generative systems for music mixing [21]. Finally, neurofeedback training systems have been developed in the effort to enhance music and creative performance [8].

2.2 Electro-physiology Sensor Systems

Conventional electro-physiology systems use electrical conductors to measure electrical signal derived from brain and body activity. For instance, Brain Computer Interfaces (BCI) use electrodes placed in the scalp to measure brain electrical activity (EEG) and transform it into commands that allow control of devices and applications [23]. Therefore, it provides a non-muscular communication channel that has been widely used in clinical rehabilitation. Physiological interfaces may also include the measurement of other biopotentials different to brainwaves, such as electrocardiography (ECG) or electrooculography (EOG), using a single device.

2.3 Musical Tabletop Interfaces

There has been a proliferation of musical tabletops in the past decade. Projects such as the Audiopad [22], the MusicTable [2] or the Reactable [13], started showing the possibilities and affordances of tangible tabletop musical instruments. Some of these devices are more oriented towards sound synthesis (e.g. Reactable), some towards composition (e.g. Xenakis [3]) or sequencing (e.g. Scrapple [18]). Some are meant for professional or experienced musicians, while others are more oriented towards education or entertainment (e.g. Zen Waves [7]).

Independently of the many differences that can exist between all these systems, scholars tend to agree in the benefits resulting from interacting with these large-scale tangible and multi-touch devices. Their vast screens make them excellent candidates for collaborative interaction and shared control [6], while supporting real-time, multidimensional as well as explorative interaction. These characteristics also make tabletops especially suited for both novice and expert users. Additionally, we think that the visual feedback possibilities of this type of interfaces, makes them ideal for understanding and monitoring complex mechanisms, such as the several simultaneous processes that can take place in a digital system for music performance [13].

3. SYSTEM ARCHITECTURE

In this paper we present a first working prototype of a multimodal system for collaborative sound generation and control, combining physiological computing and a tabletop interface[1]. This section describes the extraction and processing of the physiological signal, the mappings applied for physiology-based sonification, its parameters for sound generation and control, finishing with the integration with the Reactable framework.

[1]Video available on `http://www.vimeo.com/14675468`

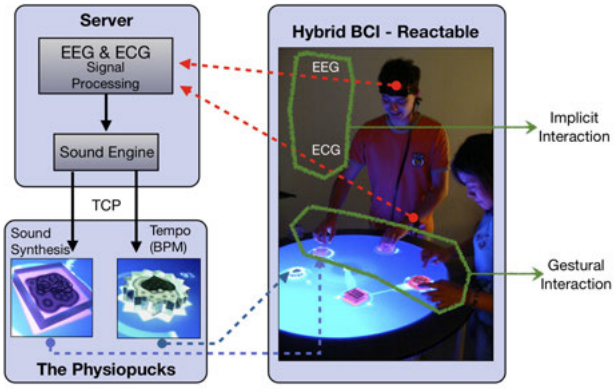


Figure 1: Multimodal Music System. Physiological signals (red dotted arrows) are wirelessly streamed to a server that applies a signal processing and sonification. EEG-based sound synthesis and tempo control through heart rate are integrated in the Reactable framework, and presented to performers as physiopucks (blue dotted arrows).

3.1 Signal Extraction and Processing

The proposed system uses Starlab's *Enobio* for physiological signal extraction. *Enobio* is a wearable, wireless electrophysiology sensor system that captures three biopotentials: EEG, ECG and EOG. It features 4 channels connected to dry active electrodes with a sample rate of 250hz, a resolution of 0.589μv, maximum Signal-to-Noise Ratio of 83db, a 16-bit Successive-Approximation Register (SAR) Analog-to-Digital Converter, and an automatic offset compensation for each channel [26].

Figure 1 describes the system's design. A dry electrode is placed on the frontal midline (Fz) lobe of participants for EEG recording [16]. The electrode for heart rate detection is placed in the wrist of subjects using a wristband. Physiological signals are acquired, amplified and streamed wirelessly to a server application for processing and sonification. There, the synchronization is managed by the Enobio software suite, that applies a digital filter to reduce noise (centered between 50 and 60hz) and sends the EEG and heart rate data to the sound engine. At this stage, a EEG-based sound synthesis and a tempo control based on heart rate are computed and streamed to the Reactable framework via a TCP/IP port.

3.2 Sound Engine

In this study, the selection of physiology sonification methods had two motivations. First, we wanted to provide feedback with minimal delay about changes from different frequency bands of EEG. Second, we aimed at easily recognizable sonification that would stand out from other sounds generated using a musical tabletop interface.

The system's sound engine uses a direct mapping between EEG alpha-theta bands (4-12Hz) and the audible sound frequency spectrum. This mapping was motivated by alpha-band neurofeedback designs [8]. This EEG processing unit appears as a sound generator puck (brain-labeled *physiopuck*) on the Reactable. On the other hand, the heart rate is mapped to another puck to control tempo or beats per minute (BPM) on the Reactable (heart-labeled *physiopuck*) (see Figure 1).

The Pure Data (Pd) computer music system [24] performs the real-time signal analysis and sound synthesis. It has been chosen due to its openness and suitability for per-

forming such tasks, and for its flexibility when defining the mappings. This software also favors a robust integration with the Reactable framework, whose sound engine has been built with Pd.

3.2.1 EEG and Heart Rate Signal Processing

The computed magnitude spectrum for each EEG frame is used to shape the spectrum of a white noise signal. Each frequency bin is then used to weight the first 128 frequency bins of a 256 bins white noise FFT. Working at 44.1 kHz for audio synthesis, a frequency range going from 0 Hz to 11025 Hz is covered, with each frequency bin taking about 86 Hz. The spectral magnitudes are equalized by weighting the chosen curve to emphasize the weaker higher frequencies. The sound resynthesis stage consists of an overlap-add of the inverse FFT of the weighted and equalized magnitude spectrum of each consecutive processed EEG signal block and is entirely handled by the Pd synthesis engine. The resynthesized audio signal is finally streamed over a TCP-IP/LAN connection to a server running the Reactable software, where the EEG-based sound synthesis and the heart rate tempo control are finally mapped to the *physiopucks*.

The heart rate signal is processed by first applying an adaptive rescaling of the system. A two-seconds sliding window (500 samples) checks for the minimum and maximum values. Therefore, the signal is normalized depending on that range. This adaptive approach compensates for the signal without losing heart rate peak resolution. Peaks in the heart rate are detected by applying a simple threshold function. A heartbeat is detected if the normalized signal is above the 40% of the normalized range. A new heartbeat is then detected only if this signal falls below 30%.

3.3 Integration into the Reactable

The Reactable's sound synthesis and control methods follow a modular approach, a prevalent model in electronic music, which is based on the interconnection of sound generators and sound processors units. In the Reactable this is achieved by relating pucks on the surface of the table, where each puck has a dedicated function for the generation, modification or control of sound. Reactable's objects can be categorized into several functional groups such as audio generators, audio filters, controllers (which provide additional control variables to any other object) or global objects (which affect the behavior of all objects within their area of influence) [13]. Each of these families is associated with a different puck shape and can have many different members, each with a distinct and human-readable symbol on the surface. Because of this modular approach, the integration of a physiological subsystem into the standard Reactable was straighforward. Two new pucks (*physiopucks*) were created, allowing the performers to use their physiological signals to generate and control sound, in the same manner as using standard Reactable objects (see Figure 1).

4. EXPERIMENT

To assess the effect of physiology-based interaction on collaborative music experiences, and to evaluate the performance of the proposed multimodal system, we designed a task-oriented experiment of music creation involving two participants. Each experiment took around 45 minutes and was designed to measure *performance* and *motivation* using self-reported ratings. The experiment was conducted in accordance with the ethical standards laid down in the 1964 Declaration of Helsinki.

4.1 Experimental Setup and Task design

The experiment involved a pair of participants with two distinct roles: one termed *user* who operated the Reactable pucks with her hands, and one termed *emitter* who manipulated the standard pucks but also provided the physiological signals for the *physiopucks*.

These *user-emitter* pairs worked with a set of six standard Reactable pucks plus the two *physiopucks*. After a first explorative phase, two five-minute tasks were to be completed. Each task consisted in replicating 15-seconds prerecorded music excerpt created with the same pucks that were available to the participants during the test. All *user-emitter* pairs listened to the same music reference. Once the excerpt was played, the *user-emitter* pair had up to 5 minutes to mimic the sound. The participants were able to replay the reference at any time by asking the experiment leader. This task-oriented design was applied to encourage the *user-emitter* pair in a music composition process.

During the task, the *user* manipulated the pucks in the surface of the Reactable (gestural interaction) whereas the *emitter* performed both gesturally and through her own physiological signals mapped to the *physiopucks* (implicit interaction). *Physiopucks* were available for both *emitters* and *users* to be combined with any of the standard Reactable objects.

4.2 Sample and Groups

A total of 32 participants, age mean of 28.09 years old (SD=3.5), 15 females and 17 males, with no experience using the Reactable, took part in the experiment. They were distributed in two groups: a Physio Group (N = 22) where signals from the *emitter* were mapped in real-time to the *physiopucks*; and a Placebo Group (control group, N = 10) where *physiopucks* were driven by pre-recorded EEG and heart rate signals, thus providing no real feedback to the *user-emitter* pairs. Participants were unaware of this effect and *emitters* in both groups were told they were controlling the *physiopucks*. The physiological signals used by the Placebo Group were recorded from a person who composed the reference music excerpt and were similar to the ones in Physio Group.

4.3 Measures

Measures were taken using three self-reported tools: (1) Pre-test questionnaire: demographics, general music knowledge, electronic music skills and Reactable knowledge; (2) Post-test questionnaire with 10 measures representing motivation and performance, based on a 5-points Likert scale ranging from "strongly agre" to "strongly disagree", except where noted or implied; (3) Self-Assessment Manikin (SAM) using 9-points pictorial scale for emotional valence and arousal [17].

Each measure in the post-test questionnaire contained from 2 to 5 questions. The measures concerning collaborative performance were based on [12] and involve Feedback (M1), Distribution of Control (M2), Social Affinity (M3) and Nature of the Task (M4). The motivation measures were based on [11], which describes Curiousity (M5), Difficulty (M6), Confidence (M7) (10-points Likert scale), Control of the Interface (M8), Motivation (M9) and Satisfaction (M10). The detailed description of these factors and questionnaires can be found in [19].

5. RESULTS & DISCUSSION

The ratings from the abovementioned questionnaires were collected and analyzed for 4 types of participants: physio-emitters (subjects manipulating pucks and providing physi-

ological signals for the *physiopucks*); physio-users (sub manipulating pucks and interacting with physio-emitt placebo-emitters (subjects manipulating pucks, belie they were providing physiological signals for the *physio* when those were actually pre-recorded); and placebo-u (subjects manipulating pucks and interacting with plac emitters). The data was collected through computer-b questionnaires, and mean was taken over the questions responding to each measure.

Two analyses were done. First, t-tests were applie compare the means between participants within eacl perimental group (subsection 5.1) and between them (section 5.2). Second, the variation of all responses w each tested pair (emitter and user) was evaluated by plying a Pearson correlation analysis (subsection 5.3). significant differences were found for the demographic collected in the pre-test questionnaires.

5.1 Emitters vs. Users Analysis

In these analyses, we compared the differences between emitters and users within each experimental group.

5.1.1 Physio Group: Emitters vs. Users

In this analysis only *Motivation* ratings (M9) was close to significant, $t(21) = -1.90, p = .071$, with physio-emitters being more motivated than physio-users. Both types of participants reached similar levels of *Difficulty* (M6), and the *Distribution of Control* (M2) did not show significant difference between physio-emitters and physio-users (see Figure 2, left quadrants).

The lack of significant difference for all measures could be an indicator that both *emitters* and *users* within the Physio Group had a similar experience during collaboration. Importantly, these factors differed from Placebo Group, as shown in the next subsection.

5.1.2 Placebo Group: Emitters vs. Users

The analysis showed two results. *Difficulty* ratings (M6) was significant, $t(9) = -3.57, p < .01$, with placebo-emitters declaring higher challenge ($M = 2.46, SD = 0.18$) than placebo-users ($M = 1.93, SD = 0.27$) (see Figure 2, right quadrants). This may show that placebo-emitters could perceive that the feedback was not working properly. Secondly, the analysis unveiled significant differences for *Distribution of Control* (M2) ($t(9) = -2.35, p < .05$) as shown in Figure 2. Placebo Group showed an asymmetric tendency, with placebo-emitters declaring higher Control ($M = 2.80, SD = 0.44$) than placebo-users ($M = 1.80, SD = 0.83$).

A high perception of *Difficulty* from the placebo-emitters would potentially force them to take a more active role in "making system work", forcing placebo-users to give up a more active role in the control distribution.

5.2 Between Group Analyses

In these analyses, we compared ratings of emitters and users from different experimental groups.

5.2.1 Physio-Users vs. Placebo-Users

Physio-users declared higher *Confidence* (M7) ($M = 5.06$, $SD = 1.45$) in the task as compared to placebo-users ($M = 3.55, SD = 1.19$), $t(15) = 2.03, p < .05$. Importantly, while the settings were identical for *users* in both groups, the confidence of placebo-users could be affected by the lack of clear feedback perceived by placebo-emitters. In a similar manner, the difference in *Distribution of Control* (M2) was significant between users in both groups, $t(15) = 2.6, p < .05$. Operating under the same conditions, physio-users reported

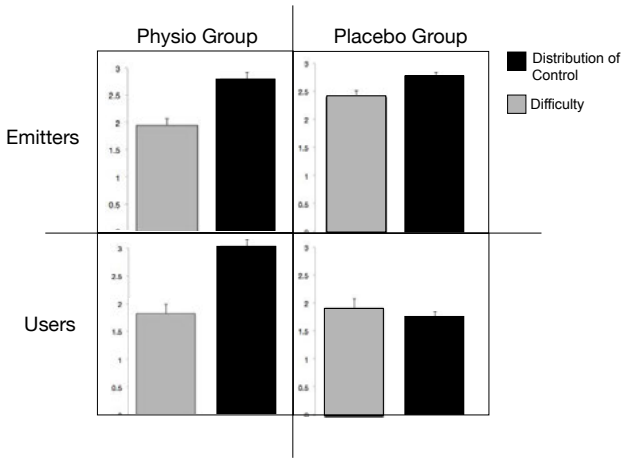


Figure 2: Ratings for *Difficulty* (M6) and *Distribution of Control* (M2) measures in four participant types (scale from 0 to 5). Error bars show standard deviation. See sections 5.1 and 5.2 for the details.

higher *Control* ($M = 3.00, SD = 0.89$) than placebo-users ($M = 1.80, SD = 0.83$). Figure 2 (lower quadrants) clearly shows this effect. As mentioned in section 5.1.1, physio-users did not show an asymmetric *Distribution of Control* compared to their physio-emitters. However, this measure is significantly lower for placebo-users compared to placebo-emitters.

5.2.2 Physio-Emitters vs. Placebo-Emitters

Physio-emitters showed higher *Confidence* levels (M7), ($M = 4.90, SD = 1.06$) compared to placebo-emitters ($M = 3.65, SD = 0.96$) at $t(15) = 2.24, p < .05$. Second, placebo-emitters reported greater *Difficulty* (M6)($M = 2.46; SD = 0.18$) than physio-emitters ($M = 1.96, SD = 0.64$) $t(15) = -2.37, p < .05$. The introduction of a sham pre-recorded signal for placebo-emitter had a clear effect not only in the performance and motivation of these participants, but also in the role of their partners (i.e. placebo-users).

5.3 Correlation analysis

To study in depth the synchronization between *user* and *emitter* in participant's pair, we applied a correlation analysis to evaluate the consistency between their responses to each questions. When all measures were combined together, both Physio and Placebo groups show high level of response consistency between user-emitter pairs. Interestingly, when correlations were analyzed measure by measure, a different picture emerged (see Table 1).

The *Feedback* measure (M1) showed higher correlation for Physio pairs ($r = 0.51$) than for Placebo pairs ($r = 0.25$). In the case of the former, the correlation level shows the importance that both participants assigned to the audiovisual feedback coming from the system during the collaborative tasks. Placebo pairs responses are almost uncorrelated, which indicates that placebo-emitters were not able to recognize the feedback coming from the Reactable, and such a factor also affected placebo-users collaborating with them.

The correlation analysis of *Collaborative nature of the tasks* (M4) showed differences between Physio and Placebo user-emitter pairs. Whereas the Physio pairs showed moderate and significant correlation between participants (i.e. there was an agreement on considering the tasks as collaborative), the Placebo pairs' ratings were not correlated. This

Table 1: Pearson correlation coefficients of *user-emitter* pairs responses for Physio and Placebo Groups (*significance at 0.05, **at 0.01, ***at 0.005 level)

Measures	Physio	Placebo
All measures	0.80***	0.68***
Feedback (M1)	0.51**	0.25
Distribution of Control (M2)	0.51	0.53
Social Affinity (M3)	0.52**	0.64
Nature of Task (M4)	0.41**	0.21
Curiousity (M5)	0.49***	0.63*
Difficulty (M6)	0.43*	0.72*
Confidence (M7)	0.81***	0.51***
Control of the Interface (M8)	0.11	0.42
Motivation (M9)	0.34	0.03
Satisfaction (M10)	0.62***	0.62**
Arousal	0.25	0.6
Valence	0.13	-0.17

result supports the feasibility of physiology-based interaction for music collaboration.

For *Difficulty* measure (M6), ratings from user-emitter pairs were highly correlated in Placebo but not for Physio group (see Figure 1). Interestingly, valence-arousal ratings were not highly correlated except arousal ratings for Placebo group, which corroborates the results for difficulty measure.

Measure of *Control of the Interface* (M8) showed moderate correlation for Placebo, but not for Physio Group. Together with a significant asymmetry between emitters and users in the Placebo Group when running the t-test, this shows that this asymmetry was consistent among its *user-emitter* pairs.

Finally, *Motivation* measure (M9) were almost uncorrelated between *user-emitter* pairs in both groups. This is especially interesting for the Placebo Group, as it shows a tendency to lose interest in the performance during collaboration.

6. GENERAL DISCUSSION

The presented results highlight specific aspects of a system that combines implicit, physiology-based and explicit, tabletop-based interaction in music collaboration. Similar levels of rated difficulty and strong correlation of confidence ratings for user-emitter pairs in Physiology group show that this new multimodal system do not impose major difficulties for music collaboration. On the contrary, the similar ratings of distribution of control - a fundamental factor for assessing the symmetry of music collaboration - that were given by the Physio Group (but not Placebo) show that the proposed implicit interaction model encouraged symmetric music collaboration between the participants.

The results also show that placebo-emitters expressed higher levels of difficulty and lower levels of confidence. While such experiences were expected for participants who where provided with a fake biofeedback, it is notable how these affected the experience of their partners, placebo-users. As an example, we can mention the significantly lower level of confidence in placebo-users as compared to physio-users, regardless them both operating the system in the same conditions. This reciprocity effect in the performance of participants has to be taken into account in the design of multimodal interfaces for music collaboration.

The experiment also helps to understand the perceptual aspects of display techniques based on physiological signals. The scores corresponding to audiovisual feedback reached a high correlation in the Physio Group, but not in the Placebo Group. This indicates that the participants were able to perceive whether the feedback from the sonification engine and the Reactable graphical interface was linked to their physiological signals or not. This factor is particularly interesting for collaborative music performances, as it shows that a direct mapping between EEG spectral bands and the audible sound frequency spectrum is effective as an identifiable auditory display. It also unveils that both *emitters* and *users* were able to recognize the sound processes driven by physiological signals, within a multimodal musical interface that included other control paradigms (e.g. gestural input). However, the musical expressivity arising from such design has to be further explored, as discussed in the next section.

6.1 Future Work

Several potential upgrades for the system are foreseen. First, alternative EEG sensing devices can be used in order to improve signal acquisition and cover other regions of the brain. Second, regardless the fact that subjects did not perceive significant latency when running the experiment, a better communication protocol can be applied to improve the connectivity between modules and reduce latency, for instance by using Open Sound Control (OSC). Finally, other sonification mappings can be applied in order to achieve higher musical expressiveness and intuitiveness. Designs based on adaptive systems can be envisioned, where physiological signals are monitored only covertly, in absence of user's intentional control. Such collaborative system could then passively monitor performers' perceptual, cognitive and emotional states and use real-time machine learning methods for adaptive multisensory feedback. [5] [15] [14].

Future experiments can be complemented with time measures (e.g., how long does it take to complete a task using the system), physiological measures (recording of EEG, ECG and EDA) that characterize psychophysiological states, visual recording for behavioral observation (gestures, facial expressions), qualitative data from the participants and similarity metrics between the sound references and recorded trials. Importantly, future studies will involve pairs of *emitters* performing together, instead of a *user-emitter* design. This will allow to study physiological synchronization between performers. Finally, to assess the musical possibilities of the multimodal system, experiments with professional musicians can be carried on, given their previous training.

7. CONCLUSIONS

Physiological computing in collaborative HCI applications is a rapidly developing field of research that require new experimental paradigms and methodologies. This paper presents a multimodal system for music collaboration, and a methodology for assessing participants' performance and motivation. The analysis has shown that the combination of implicit, physiology-based and explicit, tangible interaction is (a) feasible for participants collaborating in music composition, and (b) that it preserves a balanced distribution of control between collaborators. These results strongly support the use of physiological interfaces for music collaboration, as they can lead to meaningful and novel experiences in the field of CSCW and music creation. Together with the creation and control of sounds, brain and body signals may be powerful indicators of performer's emotional and cognitive states during collaboration, guiding music anticipation and interpersonal synchronization.

8. ACKNOWLEDGMENTS

This work was supported by TEC2010-11599-E and MAEC-AECID. We want to thank Arnau Espinoza, Carles F. Julià, Daniel Gallardo and Eliza-Nefeli Tsaoussi for their advice.

9. REFERENCES

[1] J. Allanson and S. Fairclough. A research agenda for physiological computing. *Interacting with Computers*, 16(5):857–878, october 2004.

[2] R. Berry, M. Makino, N. Hikawa, and M. Suzuki. The augmented composer project: The music table. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 338. IEEE Computer Society, 2003.

[3] M. Bischof, B. Conradi, P. Lachenmaier, K. Linde, M. Meier, P. Pötzl, and E. André. Combining tangible interaction with probability-based musical composition. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 121–124. ACM, 2008.

[4] G. Edlinger, C. Holzner, C. Groenegress, C. Guger, and M. Slater. Goal-oriented control with brain-computer interface. In D. Schmorrow, I. Estabrooke, and M. Grootjen, editors, *Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience*, pages 732–740. Springer Berlin / Heidelberg, 2009.

[5] S. H. Fairclough, K. M. Gilleade, L. E. Nacke, and R. L. Mandryk. Brain and body interfaces : Designing for meaningful interaction. In *CHI 2011*, pages 1–4, 2011.

[6] Y. Fernaeus, J. Tholander, and M. Jonsson. Beyond representations: towards an action-centric perspective on tangible interaction. *International Journal of Arts and Technology*, 1(3):249–267, 2008.

[7] E. Glinert, N. Kakikuchi, J. Furtado, T. Wang, and B. Howel. Zen waves. Tangible Media Group, MIT Media Lab, Boston, 2008.

[8] J. Gruzelier. A theory of alpha/theta neurofeedback, creative performance enhancement, long distance functional connectivity and psychological integration. *Journal of Neurophysiology*, 10:101–9, 2009.

[9] T. Hermann, T. Bovermann, E. Riedenklau, and H. Ritter. Tangible computing for interactive sonification of multivariate data. In *2nd International Workshop on Interactive Sonification*, pages 1–5, York, UK, 2007.

[10] T. Hermann and A. Hunt. The discipline of interactive sonification. *Proceedings of the Int. Workshop on Interactive Sonification*, pages 1–9, 2004.

[11] K. Issroff and T. del Soldato. Incorporating motivation into computer-supported collaborative learning. In *Proceedings of European conference on artificial intelligence in education*, 2006.

[12] A. Jones and K. Issroff. Learning technologies: Affective and social issues in computer-supported collaborative learning. *Computers & Education*, 44(4):395–408, 2005.

[13] S. Jordà. On stage: the reactable and other musical tangibles go real. *International Journal of Arts and Technology*, 1(3/4):268–287, 2008.

[14] M. Kaipainen, N. Ravaja, P. Tikka, R. Vuori, R. Pugliese, and M. Rapino. Enactive Systems and Enactive Media. Embodied human - machine coupling beyond interfaces. *Leonardo*, 5(44), 2011.

[15] A. Y. Kaplan, J.-G. Byeon, J.-J. Lim, K.-S. Jin, and B.-W. Park. Unconscious Operant Conditioning in the Paradigm of Brain-Computer Interface Based on Color Perception. *Intern. J. Neuroscience*, 115(1):781–802, 2005.

[16] J. D. Kropotov. *Quantitative EEG, Event Related Potentials and Neurotherapy*. Academic Press, San Diego, CA, 2009.

[17] P. J. Lang. Behavioral treatment and bio-behavioral assessment: computer applications. In J. J. J.B Sidowski and T.A.Williams, editors, *Technology in Mental Health Care Delivery Systems*, pages 119–137. 2005.

[18] G. Levini. The table is the score: An augmented-reality interface for real-time, tangible, spectrographic performance. In *Proceedings of ICMC*. School of Art, Carnegie Mellon University, 2006.

[19] S. Mealla. Effects of physiology-based interaction in collaborative experiences, 2010.

[20] E. Miranda, S. Roberts, and M. Stokes. On generating eeg for controlling musical systems. *Biomedizinische Technik*, 49(1):75–76, 2004.

[21] E. R. Miranda and V. Soucaret. Mix-it-yourself with a brain-computer music interface. In *ICDVRAT*, pages 1–7, Maia, Portugal, 2008.

[22] J. Patten, B. Recht, and H. Ishii. Audiopad: a tag-based interface for musical performance. In *Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6. National University of Singapore, 2002.

[23] G. Pfurtscheller, B. Z. Allison, C. Brunner, G. Bauernfeind, T. Solis-Escalante, R. Scherer, T. O. Zander, G. Mueller-Putz, C. Neuper, and N. Birbaumer. The Hybrid BCI. *Frontiers in neuroscience*, 4(April):42, 2010.

[24] M. Puckette. Max at seventeen. *Computer Music Journal*, 24(4):31–43, Winter 2002.

[25] D. Rosenboom. Extended musical interface with the human nervous system. *Leonardo Monograph Series. International Society for the Arts, Sciences and Technology (ISAST)*, (1), 1997.

[26] Starlab. *Enobio User Guide*. Starlab, Barcelona, 2010.

[27] J. Wolpaw, N. Birbaumer, D. McFarland, G. Pfurtscheller, and T. Vaughan. Brain-computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[28] B. F. Yuksel, M. Donnerer, J. Tompkin, and A. Steed. A novel brain-computer interface using a multi-touch surface. *Proceedings of the 28th international conference on Human factors in Computing Systems - CHI '10*, page 855, 2010.

Examining How Musicians Create Augmented Musical Instruments

Dan Newton and Mark T. Marshall
Interaction and Graphics Group, Department of Computer Science, University of Bristol, UK.
djslylogic@gmail.com, mark@cs.bris.ac.uk

ABSTRACT

This paper examines the creation of augmented musical instruments by a number of musicians. Equipped with a system called the Augmentalist, 10 musicians created new augmented instruments based on their traditional acoustic or electric instruments. This paper discusses the ways in which the musicians augmented their instruments, examines the similarities and differences between the resulting instruments and presents a number of interesting findings resulting from this process.

Keywords

Augmented Instruments, Instrument Design, Digital Musical Instruments, Performance

1. INTRODUCTION

Augmented musical instruments are created by the addition of sensors to existing acoustic or electric instruments. These sensors allow the performer to control additional digital audio effects or sound synthesis processes through their gestures. Such instruments offer numerous possibilities for musical performance [5], but also create issues with regard to the musicians' ability to control these extra effects [2].

Based on the idea that musicians themselves would best know how to augment their musical instruments, both in terms of gesture potential and cognitive load, we created the Augmentalist [6]. The Augmentalist is a system to allow performers to easily augment their musical instruments. It consists of a combination of hardware (sensors and a sensor interface) and an easy to use mapping software.

As part of the design process for the Augmentalist system we worked in collaboration with 10 musicians, developing the system in an iterative user-centred manner. This resulted not only in a robust and easy to use system, but also a number of new augmented instruments developed by these performers over the course of the project.

This paper details the results of this process. We begin with an overview of the Augmentalist system itself, to allow for a better understanding of how the system works and how it could be used.

2. THE AUGMENTALIST

NIME'11, 30 May–1 June 2011, Oslo, Norway.

The design process for the Augmentalist took an iterative, user-centred approach. This involved numerous consultation, testing and design sessions with musicians. The overall goal of this process was to ensure that the system was useful to the musicians themselves.

The consultation process began as soon as the project itself was conceived. It began with a series of short meetings with a number of different musicians. These meetings included sessions with single musicians and also with groups of musicians. The aim of these initial sessions was simply to gauge interest in the project itself and to attempt to determine what features of the system would be useful to a variety of musicians.

From these sessions, we arrived at a design that allowed the musicians to attach sensors the their instruments and then map the output from these sensors to MIDI signals. These MIDI signals could then be used to control parameters of audio software with which the musicians are familiar. The remainder of this section presents an overview of the hardware and software implementation of the Augmentalist. More detail of the system can be found in [6].

2.1 Hardware

After experimenting with different available sensors and sensor systems, we decided to use Phidgets [3]. These sensors require no soldering or programming on behalf of the users. Thus they are ideal for a system designed for musicians due to their plug and play capabilities. The choice of the Phidgets system also allows for a large range of sensors to be available to the user, with dozens of sensors currently available from the manufacturers that plug directly in to the interface with no electronic skills required to use them.

The sensors are connected via USB 2.0 to a computer. For our initial implementation we used a 2.53GHz MacBook Pro Running OS X 10.6 using a Stanton Scratch Amp firewire audio interface for audio input/output.

2.2 Software

To convert the sensor data into MIDI signals we used the Max/MSP programming environment. This had the advantage of being easy to use, as well as being fully compatible with the Phidget sensors. It is also an environment with which some of our musicians were familiar. The interface in Max/MSP allows the user to choose which sensors to map to specific MIDI channels, as well as setting the desired input range from the sensor, output range for the MIDI channel and the mapping between them.

The software allows the user to select which sensor is mapped to to which MIDI signal using a simple graphical interface. The range of sensor values to be mapped can be selected by demonstration, with the performer moving the sensor through its desired range. The MIDI output range can also be limited to a specific range in the software.

Finally, the user can specify the mapping function used to convert sensor data to MIDI data. This can be selected from a range of presets (linear increasing, logarithmic increasing, linear decreasing, etc.) or by drawing a mapping function in the interface.

The MIDI output could then be mapped to parameters in audio software chosen by the musicians. In our development sessions we primarily used Apple's Logic Pro 9.

3. INSTRUMENT DESIGN AND TESTING SESSIONS

Over the course of the development of the Augmentalist a group of 10 musicians spent numerous hours working with the system, creating and testing new augmented musical instruments and mappings. This group of musicians was made up of 3 guitarists, 3 bass players, 2 DJs, a saxophonist and a vocalist. Interestingly, examples of augmented instruments for each of these types of performers can already be found in the literature [5, 1, 7, 4]. This would seem to indicate that these types of musicians have the necessary spare "bandwidth" to allow them to successfully play an augmented instrument.

For the design and testing sessions, the musician was free to choose the sensors used, the attachment of the sensor to the instrument, the effect being controlled and the mapping of the sensor to the effect. This gives the musician total control over how the system is designed and used.

Each session followed the same format, as follow:

1. Presentation of software including any updated features.
2. Participant uses software with researcher present for short time researcher helps participant with any issues that arise.
3. Participant left to use software for a longer period of time.
4. Participant fills in feedback form at the end of the session.
5. Researcher performs a short interview of participant to gather any additional thought, problems etc.

The aim of this session format was to allow us to inform the users of new developments in the system and to receive as much feedback from the users as possible, without causing them to feel under pressure. The solo portion of the session, in which the participant used the system without supervision, was designed to allow them to explore the system with as much freedom as possible, and without the pressure of having an audience that could arise from our presence.

4. DEVELOPED INSTRUMENTS

In this section, we discuss the instruments and mappings developed by the musicians. In particular we look at the choices of sensors and gestures that the performers used. Each participant worked with the system for multiple 1 hour sessions over the course of the development. Each developed their own instruments and mappings. This allowed us to look for similarities between the instruments developed by different performers based on the same instrument, as well as across instruments.

4.1 Guitarists

4.1.1 Gestures and Sensing

Most interestingly, we found that all 3 guitarist used a tilting of the guitar body as a control gesture. This gesture was sensed using an accelerometer, mounted to either the body or the headstock of the guitar, depending on the performer.

One guitarist used the position of the picking hand over the guitar body as a control gesture. This was sensed using a slider mounted to the guitar body, below and parallel to the strings, as shown in Figure 1.

Figure 1: An example of a guitar augmented using the Augmentalist system.

Another interesting gesture/sensor combination that was developed involved the use of an infrared distance sensor to detect strumming rhythm. A number of possibilities for detecting strumming rhythm were discussed by the guitarists, including attaching an accelerometer to the performers strumming hand, and trying to determine strumming rhythm from the sound output. However, one guitarist decided to detect strumming rhythm using an infrared distance sensor, which was mounted on the body of the guitar, under the strings. This sensor was set up so that when the guitarist strummed the strings their hand would pass over it. It was then configured as an on/off switch which triggered whenever the guitarist's hand passed over it. This switch between on and off then provided a measurement of the strumming rhythm.

One other possibility that guitarists examined for control gestures was the use of head and body movements. Suggestions included the use of head mounted accelerometers to detect head tilting and the use of accelerometers on the body to detect weight shifting. However, these were found to be too cumbersome and/or restrictive for use when playing.

4.1.2 Audio Effects and Mapping

For each of the guitarists, the control gestures just described were mapped to a number of audio effects in Apple's Logic Pro. The choice of gestures, effects and mappings were left to the individual guitarists. Logic Pro was chosen as it is a software package that many of the participants were familiar with and also offers a large number of possible effects to control.

All of the guitarists chose to use effects that they were already somewhat familiar with and that are commonly used by guitarists playing electric guitar. These effects included distortion, delay, chorus, flanger, and master volume.

Example mappings included the control of delay using the tilt of the guitar, controlling distortion using the picking position and mapping strumming rhythm to master volume.

4.2 Bassists

4.2.1 Gestures and Sensing

The sensing of the gestures for the bass was similar to the guitar. In particular, both bassists also used the tilt of the bass guitar as a control parameter, again detected using an accelerometer mounted to the headstock or body of the instrument.

One of the bassists also tried to use body movements as a control parameter. As with the guitarists, he attempted to use head tilt (detected with an accelerometer on the head) as a control. While finding this somewhat difficult to control, he also found it extremely enjoyable and kept it as part of his instrument.

4.2.2 Audio Effects and Mapping

The effect that the bassists had the most fun with was the wah effect mapped to the accelerometer measuring the tilt of the neck. This is essentially a bi-pass cutoff where the cutoff frequency is set by the sensor. The wah has existed for many years as a foot pedal for guitarists and bassists alike but transferring this concept to the angle of the neck proved quite difficult for one bassist who had little experience with effects. Instead he ended up playing the bass as normal, with a few slight body movements in time with the music. This created very subtle changes to the ambience of the bass as the wah moved in time with the music.

Other effects tested by the bassists included distortion and filter effects. These were often mapped to the tilt of the instrument, allowing a subtle, graduated control of the effect.

4.3 DJs

4.3.1 Sensing and Gestures

Although a DJ tends to have their hands full much of the time, we found that the DJs preferred to use their hands to control the sensors, rather than finding some unused performance gesture. This meant that they were often simply utilising the properties of the sensor directly, rather than attempting to use the sensor to sense a gesture. As such the sensors often became extra controls for their mixer.

In testing, one DJ who used the system made extensive use of 3 sensors: an accelerometer, a slider and a force sensor. The accelerometer was attached to the performer's hand and used as a tilt sensor. This allowed them to control effects by tilting their hand in 2 axes. This was the only sensor which the DJ used to sense movement, rather than as a direct control.

Sliders are extremely common in DJ equipment and are used for volume, turntable speed, as well as many effects. The DJ quickly picked up on the advantage of the slider. Retaining its position and it's location next to the pitch control on the turntable allowed for quick and easy adjustments whilst mixing. When a DJ mixes, a large proportion of his time is spent focusing on the pitch control which is located on the turntable next to the tone arm and so the DJ was able to quickly switch to this slider to control effects.

The force sensor was attached to the opposite turntable in the same place. The DJ activates the force sensor by pressing on it. The force of this pressing is then measured by the sensor. Figure 2 shows the system in use.

The second DJ made use entirely of sliders, using them as additional effect controls on top of their turntable decks.

4.3.2 Audio Effects and Mapping

DJs can make use of a large number of effects during a performance, switching effects during a track or when changing tracks. As one of our DJs made use only of sliders as additional effect controls, the mapping is not particularly interesting. As such, we will concentrate on the other DJ who made use of a number of sensors and effects.

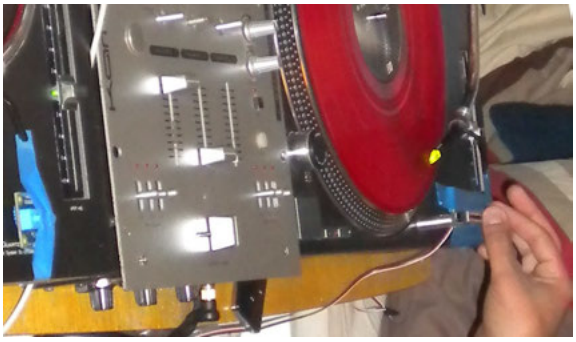


Figure 2: DJ playing with a slider and force sensor on the turntables

This DJ who tested the system tried a large number of effects with each sensor, before settling on several options for each sensor. The result is that for each sensor the DJ has a number of effects, which can be controlled one at a time or even several at once.

For the accelerometer, the effects chosen were a bandpass filter effect and a beat repeating effect called Beatmasher. While the bandpass filter resulted in controlled, predictable effects, controlling the Beatmasher resulted in interesting but more random results. Interestingly the DJs found that the Beatmasher effect was more useful when used on Techno music than Drum and Bass.

For the slider, the DJ chose the distortion drive level and the Transpose Stretch effect, which pitch shifts and time stretches the audio.

Finally, for the force sensor the DJ chose to control a number of effect mix levels, including reverb mix, delay mix, flanger mix and phaser mix. The nature of the force sensor, which returns to a zero value output as soon as the performer stops pressing on it, allowed the DJ to add and effect by pressing the sensor, increasing the effect by increasing pressure and then instantly stop the effect by releasing the sensor.

4.4 Vocalist

4.4.1 Sensing and Gestures

The vocalist (an MC who 'rapped' rather than sang) made use of hand gestures to control effects. This included sensing of the tilting of his hand in two dimensions. This was accomplished through the use an accelerometer strapped to the back of his hand.

As with the other participants, the vocalist also considered the use of head movements, again sensing head tilt using an accelerometer. However, these movements were found to be too disconcerting to use in performance.

The vocalist also examined the augmentation of the microphone. Gestures used included the sliding of the hand along the microphone (measured using a slider attached to the microphone body) and grip pressure on the microphone, detected using an FSR attached to the microphone body. Most MCs hold the microphone to perform, instead of using a stand and so to put controls on the microphone itself proved to be intuitive. Furthermore, by mapping the tightness of the grip on the microphone to an effect mix, the mapping was a natural extension of emotive performance

as with the accelerometer on the guitar.

4.4.2 Audio Effects and Mapping

This particular vocalist was not as well versed on all the various effects and their parameters as, for example, the guitarist. This meant that often effects were discovered by accident as more experimentation took place, rather than attempting to achieve a specific sound.

Something that the vocalist was keen to try straight away was a pitch shifter. This effect simply changes the pitch of the input by an amount specified on a discrete bidirectional scale. After trying with the accelerometer and struggling to maintain a steady hand (i.e. keep the pitch normal) he requested that we be able to limit the MIDI output at half so the he could keep the sensor at 0 more easily. After this he found it very intuitive to map a drop in pitch to the downwards movement of his hand and keep the pitch at 0 with his hand up.

The pressure sensor with its 'return to zero' style of operation worked really well with effects that made the sound messy as when released the effect would return to normal. Delay mix and reverb mix as well a flanger intensity worked well to accent and in some cases twist quite dramatically the sound before snapping back to a dry signal when released.

4.5 Saxophonist

4.5.1 Sensing and Gestures

In a similar way to the DJs that worked with the system, the saxophonist talked more about the sensors as extra controls rather than a medium for interpreting gestures. Perhaps influenced by his familiarity with studio sound equipment (this saxophonist was also a keen producer of electronic music), the first sensor he chose to use was the slider. A slider mounted on the saxophone body was used to control a variety of effects.

A more interesting idea that came out of his sessions was utilising the free thumb of the right hand to control effects. The saxophone has a thumb rest for holding the instrument as show in Figure 3. By placing a force sensor around this area, the saxophonist was able to squeeze the saxophone with his spare thumb to invoke an effect.

Figure 3: The thumb rest on a saxophone

The saxophonist also experimented with the use of an accelerometer to measure tilting of the saxophone in the vertical plane and the placement of a slider on his body at the hip.

4.5.2 Audio Effects and Mapping

As previously mentioned this saxophonist was also a music producer, and as such experimented with more intriguing and less popular effect parameters. One of the benefits of the Augmentalist system is its generality and the way that you can map to virtually any parameter. This allows you to add controls to parameters that you would not normally move during performance. The saxophonist was keen to explore these possibilities, and although they did not always work as expected, they were always interesting.

One effect that appeared to work was a tape delay time. The amount of time that a delay takes to repeat is usually not moved, or only moved by small amounts to keep it in time. However when used aggressively by the saxophonist, some unusual and interesting sonic effects were created. Other effects looked at included a reverb effect and a number of filter effects.

While the saxophonist experimented with all of these effects using both the slider and the thumb rest pressure sensor, we found that he significantly preferred to use slider as the main control. From interviews we determined that this was due to a combination of the "return to zero" nature of the FSR not allowing values to be maintained easily and the difficulty of manipulating thumb pressure while playing. While this second difficulty could be overcome with practice, the first is inherent to the sensor design.

5. DISCUSSION

This work has raised a number of issues regarding the development of augmented instruments by musicians. In this section we discuss in more detail some of the more interesting points raised during the development process and their implications for research into augmented instruments.

5.1 Similarities in Control Gestures

Most of the participants in the development process used or tried to use head movements and/or center of mass movements as a control parameter, irrespective of the instrument they played. This happened for a number of different performers including the guitarist, bassist, DJ and vocalist. It seems that these movements are considered by many people to be useful as additional controls in instrumental performance. Interestingly however, while most of the participants tried to use these gestures as controls, after some practice only one of them kept these gestures. This may indicate that what seems to be the most naturally useful gestures are not so useful in practice.

We also found that tilting the instrument was a commonly used control. The guitarist, bassists and saxophonist all made use of this gesture. This gesture has also been used with other wind instruments by other researchers and performers [10, 7], which would seem to indicate that this is a generally useful gesture for many instrumental performers.

5.2 Musicians as developers

The Augmentalist system was designed from the start to allow musicians to become the developers of their own augmented instruments. We believe that it is musicians who know the most about their instruments and about the sounds and music they wish to create and so it is the musician who should make the decisions on how the instrument should work.

Over the course of the development of this system so far, the participating musicians developed hundreds of different gesture to sound mappings. While this is a large number of different mappings, what is interesting is that the musicians themselves considered far more of these mappings to be successful than not. This paper has covered only a selection of those that were considered best by the performers and that they continued to use across multiple sessions. The system not only enabled them to develop new mappings, but re-

sulted in mappings the musicians found to be interesting, useful and musical.

The use of the Augmentalist also resulted in some interesting discoveries. Most notable is the use of the infrared distance sensor as a switch to detect strumming by one guitarist. This provides a very simple and robust method of detecting strumming, and is one that we have been not previously seen in the literature on augmented instruments. Secondly, we found that one of the musicians developed a mapping for their guitar that mirrored that presented in [5] and did so within the first 2 hours of using the system. This emphasises how quickly interesting and usable instruments can be developed by a musician when given access to such a system.

5.3 Focus on technology

Our initial idea of how musicians would develop their instruments was that they would first decide on a gesture to detect and a sonic output to control with that gesture, and then on how to detect this gesture. However, over the course of this project we found that many of the musicians instead focused on the technology itself. They started by examining the sensors that were available to them, the parameters these sensors could detect and where on the instrument they could be easily mounted. Only then would they think about the gestures that the sensors could be used for.

Such a focus on the technology is a somewhat interesting finding. It seems that the musicians consider the sensor technology to be the weakest link in the system and so allow themselves to be guided by the limitations of the sensors. While musicians' creativity can often thrive off such boundaries and limitations [8], if we wish to develop a system that truly focuses on the gestures and sounds then we must alter the users' perception of such limitations of the sensors.

5.4 Potential for exploration and mastery

Wessel and Wright state that a goal for designing new digital musical instruments should be for them to have a "low entry fee" together with "no ceiling on virtuosity" [11]. This means that such instruments should be simple to begin playing, but complex and engaging enough to offer the possibility of exploration and mastery.

One of the advantages of augmented instruments is that they are based around existing musical instruments. A guitarist will still be able to play an augmented guitar as though it is a regular guitar. This makes the instrument easy to begin using. The additional sensors then extend the performance possibilities of the instrument, thus allowing for more potential for creative exploration. When speaking with the musicians who worked with the system, we found they commonly expressed the belief that they could gain full control of the instrument and make best use of the system given enough time. This shows that the Augmentalist offers potential for further exploration and creativity.

5.5 Subtle Sonic Effects

When working with the bassists that took part in this research, we noticed that they seemed to produce very musical results when the effects used were quite subtle. The mapping features of the Augmentalist software, combined with the use of sensors to detect relatively large range movements combined to allow small movements to produce very subtle effects. This resulted in effects that were subtle enough to allow the musician to focus on making music rather than making the effects. We found that when the bassists started to focus on the music and not the effect mapping the effects became more subtle and natural by virtue of not being purposefully moved. This result aligns with the work presented by Lahdeoja et al [5].

5.6 Transferability of developed instruments

We have already mentioned that one musician developed a mapping for their instrument that directly mirrored a system discussed in the literature. This indicates that there is some common pool of gestures that are suggested by the design of the instrument itself, similar to those discussed by Wanderley [9]. As such, we would also expect that it is possible to transfer developed instruments between performers that play the same instrument.

To examine this we asked another guitarist to try and perform using the mappings developed by one of the guitarists working with our system. The new guitarist found it easy to begin performing using any of these mappings. In each case, it took only a few minutes of practice before they were able to utilise the gestures in performance. The guitarist also made a number of comments on how "easy" and "natural" the mappings were to use.

It is possible therefore to transfer mappings between performers of the same instrument. This means that the potential exists to share instrument designs and mappings across users. One possibility would be the creation of a community to promote such sharing between musicians interested in developing augmented instruments. We are now beginning to investigate this possibility.

5.7 Use in Ensemble Performance

One of the goals of the Augmentalist system was to allow musicians to create new augmented instruments that they could use in their own musical performances, as part of their performance careers. While our testing and development sessions focused on working with individual musicians, we also encouraged the musicians to take their augmented instruments with them for use in both alone and in conjunction with other musicians.

In every case, the musicians were happy to continue working with their augmented instruments in private. However, one of the guitarist, a member of a 3-piece rock band, also asked to demonstrate the system to his band mates. As a result, he performed with a drummer and bassist, using his augmented guitar. All the musicians found the experience enlightening and fun, and felt that the performance was enriched because of the system. The band have stated their intention to use the system in future live performances and are currently in the process of incorporating the system into their act.

5.8 Performance Bandwidth and Practice

As discussed by [2], some musicians have "spare bandwidth" when it comes to performing. This means that for these musicians, it is possible to extend their performance technique without putting too much of a load on their capabilities and reducing the quality of their performance. For most of our musicians this seemed to be the case, with some small exceptions.

We found that the vocalist and saxophonists had difficulties with some performance gestures. Concentrating on hand movements (for the vocalist) or thumb pressure (for the saxophonist) distracted from the performance. The vocalist even found that concentrating on hand movements could result in him forgetting the lyrics. This may be a case of the vocalist having exceeded his available "bandwidth". However, in both cases we noticed some improvement with time and practice, so this may also be a problem that could be overcome in time. The effect of practice on this sort of performance will form an interesting area of further study for us.

5.9 Creativity and enjoyment

One of the main aims of this project was to produce a system that facilitated both creativity and enjoyment for the musicians using it. As has already been discussed, the musicians who used the system expressed the belief that the Augmentalist allows for new musical possibilities and offers much potential for further exploration and mastery. The question that then arises is: did the musicians enjoy using the system?

Throughout the development of the Augmentalist there were regular testing, development and performance sessions involving musicians. At the end of each session, we ask each musicians to fill out a short questionnaire, which involved rating the system on a 1-to-5 scale on a number of criteria. Perhaps the most interesting result of this was that every musician gave the system a maximum rating of 5 (Very High) for enjoyment at the end of every session.

Another interesting finding is that the average rating given by the performers to the system across all the measured criteria (ease of use, enjoyment, controllability, and expressive potential) increased over time. Figure 4 shows the mean performer rating of the system over the 9 weeks of development. As can be seen the mean rating rises from 3.3 to 4.6 over this period of time.

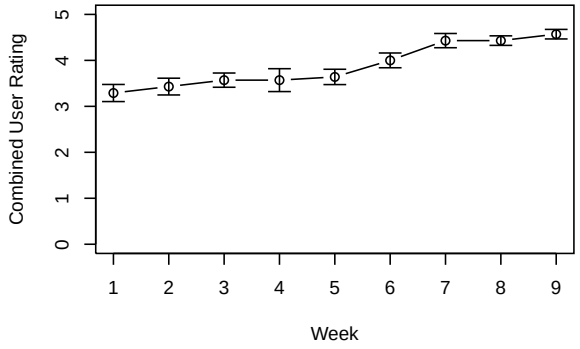


Figure 4: Performer rating of the Augmentalist system over the 9 weeks of development

While these ratings show that musicians definitely enjoyed working with the system, we think that the following quote, received from one participant several hours after a session, fully illustrates the level of enjoyment felt by those using the system:

> *"I haven't stopped smiling for ages, that was ******* awesome. When can I come back?"*

6. CONCLUSION

The main goal of the Augmentalist system was to enable musicians to begin experimenting with digital musical instruments through augmenting their existing musical instruments. Our belief was that by focusing on existing instruments and augmenting them with sensors, musicians could produce new instruments with extended interaction and performance possibilities. Such instruments would also have the advantage of reducing the performer-instrument and audience-instrument disconnect that can be present with many new digital musical instruments.

In this paper we have described a number of the instruments and mappings that our group of musicians have created using the Augmentalist system. By examining these instruments we have seen the similarities and differences between instruments designed by performers, whether playing the same instruments or difference ones. We have looked at issues such as the longer term development of these instruments by musicians, the possibility of sharing and exchanging ideas and mappings for such instruments and the innovative performance and interaction techniques that musicians develop as part of this process.

Overall, the Augmentalist allows musicians to explore new musical techniques, while also allowing them to design and create their own instruments. It opens a number of performance possibilities for these musicians and we hope in the future to be able to work with our musicians to integrate the system permanently into their performance careers, whether as soloists or as part of ensembles.

7. REFERENCES

[1] T. Beamish, K. MacLean, and S. S. Fels. Manipulating music: Multimodal interaction for djs. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 327–334, Vienna, Austria, 2004. ACM.

[2] P. Cook. Principles for designing computer music controllers. In *Proceedings of the 2001 conference on New interfaces for musical expression*, NIME '01, pages 1–4, Seattle, Washington, USA, 2001. ACM.

[3] S. Greenberg and C. Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 209–218, Orlando, Florida, USA, 2001. ACM.

[4] D. Hewitt and I. Stevenson. E-mic: extended mic-stand interface controller. In *Proceedings of the 2003 conference on New interfaces for musical expression*, NIME '03, pages 122–128, Montreal, Quebec, Canada, 2003.

[5] O. Lähdeoja, M. M. Wanderley, and J. Malloch. Instrument augmentation using ancillary gestures for subtle sonic effects. In *Proceedings of the 6th Sound and Music Computing Conference*, SMC '09, pages 327–330, Porto, Portugal, 2009.

[6] D. Newton and M. T. Marshall. The augmentalist: Enabling musicians to develop augmented musical instruments. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, pages 249–252, Funchal, Portugal, 2011. ACM.

[7] S. Schiesser and C. Traube. On making and playing an electronically-augmented saxophone. In *Proceedings of the 2006 conference on New interfaces for musical expression*, NIME '06, pages 308–313, Paris, France, 2006. IRCAM - Centre Pompidou.

[8] A. Tanaka. Musical performance practice on sensor-based instruments. In M. M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 389–405. IRCAM - Centre Pompidou, 2000.

[9] M. M. Wanderley. Quantitative analysis of non-obvious performer gestures. In I. Wachsmuth and T. Sowa, editors, *Gesture and Sign Language in Human-Computer Interaction*, Lecture Notes in Computer Science, pages 241–253. Springer Berlin / Heidelberg, 2002.

[10] M. M. Wanderley and P. Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE*, 92(4):632–644, 2004.

[11] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

Tahakum: A Multi-Purpose Audio Control Framework

Zachary Seldess
King Abdullah University of Science and Technology
Visualization Lab
Thuwal, Saudi Arabia
zachary.seldess@kaust.edu.sa

Toshiro Yamada
University of California, San Diego
California Institute for Telecommunications and Information Technology
La Jolla, CA, USA
toyamada@ucsd.edu

ABSTRACT
We present "Tahakum", an open source, extensible collection of software tools designed to enhance workflow on multi-channel audio systems within complex multi-functional research and development environments. Tahakum aims to provide critical functionality required across a broad spectrum of audio systems usage scenarios, while at the same time remaining sufficiently open as to easily support modifications and extensions via 3rd party hardware and software. Features provided in the framework include software for custom mixing/routing and audio system preset automation, software for network message routing/redirection and protocol conversion, and software for dynamic audio asset management and control.

Keywords
Audio Control Systems, Audio for VR, Max/MSP, Spatial Audio

1. INTRODUCTION
Audio Systems within interdisciplinary and multi-media research facilities are often expected to fulfill a large variety of end-user and developer functions, ranging from simpler tasks such as live event sound reinforcement and fixed media playback, to more complex activities such as experimental real-time acoustics simulations, and new musical interface design. Successfully managing audio systems in multi-functional environments relies not only on quality hardware and software implementations, but also on the existence of an overarching audio control framework. Such a framework must tackle the unique challenge of achieving a balance between stability and end-user friendliness, and flexibility and low-level access and control, ensuring successful typical daily operations, while at the same time providing a fast development pipeline.

Audio interfaces break or get replaced with better alternatives, input and output channel counts and loudspeaker configurations change over time, as do notions of ideal software and hardware solutions for all manner of lab audio functions and research projects. A successful audio control framework, in addition to facilitating smooth workflow during stable periods of operation, must also attempt to make system changes as seamless as possible during times of growth and transition.

Ideally, within a research facility, mid and high-level control of an audio system must be readily available to staff for day-to-day activities such as project demonstrations, video conferencing, and fixed-media A/V playback. At the same time, audio systems developers and technicians need to be able to experiment, modify, and implement low-level hardware and software configurations with relative ease and fluency. In this paper we present "Tahakum[1]", a set of open source, extensible software tools, designed to enhance operations and development workflow in dynamic multi-media, multi-purpose spaces.

Organization: The paper is organized as follows: In section 2 we provide an overview of the audio control and development framework, including details on our general framework design philosophy. Sections 3, 4, and 5 provide more detailed functionality and design information for AudioSwitcher Server, Control Proxy, and Asset Manager, respectively. In Section 6, we provide concluding thoughts on the framework's current state, and discuss planned and potential future improvements to the system.

2. FRAMEWORK OVERVIEW
Tahakum, created using Max/MSP, is designed to allow audio developers and technicians to easily adapt customized control systems to a given room, and to minimize the downtime in hardware and software changes within a system, while providing a baseline control framework that is easily extensible and customizable to users' equipment, preferences, and needs. Much previous work has addressed specific workflow issues within multi-functional media environments, such as spatial audio post-production (e.g. [2], [3]), real-time spatial sound rendering and composition (e.g. [4], [6], [7], [9]), and interactive room acoustics simulation (e.g. [1], [5]). Other notable work, such as [8], provides a more comprehensive framework geared specifically towards the task of sound spatialization. And there are myriad sophisticated commercial audio show control tools available, such as Meyer Sound Laboratories' CueStation[2] and Figure 53's QLab[3]. Our intention in designing the Tahakum framework has been to provide, using Max/MSP, the software tools to enhance core operational and development workflows within complex multi-media spaces, while making a point of not hindering users' preferences towards enhancements and extensions to the system via 3rd party hardware and software, network and MIDI i/o control. Our framework provides this functionality using three software tools whose primary features breakdown as follows:

[1] Tahakum, or تحكم, means "control" in Arabic.
[2] www.meyersound.com/pdf/products/lcs_series/CSv4_20070919.pdf
[3] http://figure53.com/qlab/

- **AudioSwitcher Server** combines audio mixing, routing, and delays, network and MIDI i/o, and a graphical user interface, within a preset-based automation system, for storage and recall of complex audio system state changes and event sequences. Additionally, client control software enables multi-user simultaneous access to the server.

- **Asset Manager** provides dynamic loading, unloading and control over Max patches, easy network integration using Open Sound Control (OSC), and an abstraction layer that facilitates changes to a project's panning algorithms and software-to-hardware channel mappings.

- **Control Proxy** blends network message routing/redirection, network protocol conversion between incoming and outgoing messages, and a console interface for manually sending network messages to user-defined destinations.

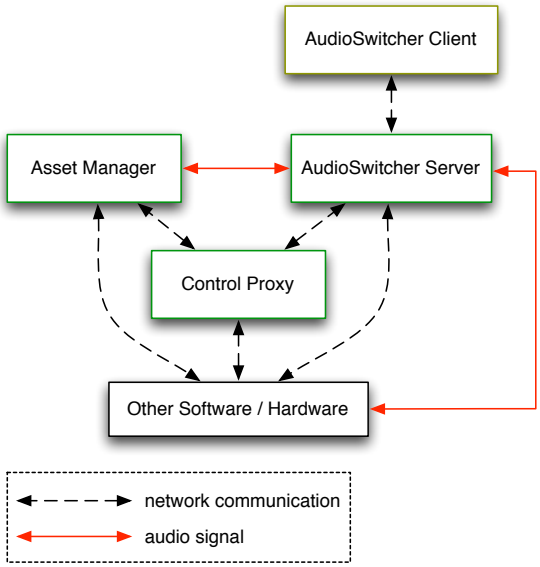


Figure 1: Typical Tahakum framework data and signal-flow

Each of the above applications is customized using plaintext configuration files with a simple syntax. By using a configuration file to control all aspects of the software's initialization and customization, the software can be updated, remotely or locally, while it is up and running. This has proven to be a significant workflow enhancement in managing our own audio systems, as well as providing an easy way to hand off new system updates to collaborators for review and discussion. Additionally, since customization is achieved via text files rather than manual Max re-patching, the software's core functionality remains the same whether running as a standalone application, with Max/MSP Runtime, or with an authorized full Max/MSP install.

All applications provide network i/o using standard protocols (including OSC) for two-way communication between each tool in the framework, as well as between various other 3rd party hardware and software products. Command syntax between applications is documented and consistent, making it possible for developers to extend the framework functionality with their own custom-designed software or hardware.

3. AUDIOSWITCHER SERVER

In this section, we give an overview of AudioSwitcher Server's functionality and briefly discuss some of the software's key controls and features.

Figure 2: AudioSwitcher Server software

3.1 Overview

Let us assume we have a multi-purpose room with the following equipment:

- 1 Blu-Ray player (with 5.1 audio output)
- 4 wireless microphones
- 1 video conferencing unit (with 2 channels i/o)
- 1 computer for custom audio (with 8 channels i/o)
- 1 computer for graphics work (with 2 channels)
- 1 hardware audio mixer with 8 channels of i/o.
- 8 loudspeakers, 1 subwoofer

Ideally, all devices in the room need to be able to send audio out to any number of the nine available speakers. Additionally, some devices, such as the video conferencing unit and the custom audio computer, need to *receive* audio from various sources as well. In many situations, it is often desirable to pass all audio through one central hub, allowing easy control over the entire system without having to physically patch cable. Assuming you have a computer with enough digital and/or analog audio inputs and outputs, AudioSwitcher Server is designed to facilitate control of idiosyncratic configurations such as the one listed above. The software is built with scalability in mind and will therefore function in a variety of scenarios, ranging from very simple to complex i/o configurations.

Summary of Functionality:

- Custom hardware-to-software i/o channel mappings
- Preset-based automation system
- Configuration files for server setup and preset definitions
- Mute/solo/delay controls on input/bus/output channels
- Input-to-bus and bus-to-output sub-mixing
- Group fader assignments on input/bus/output channels
- Control of multiple servers via client control software
- User-defined network and MIDI i/o "bindings" of server controls
- Custom network message creation, storage, and delivery to remote destinations

3.2 Signal Control, Signal Flow

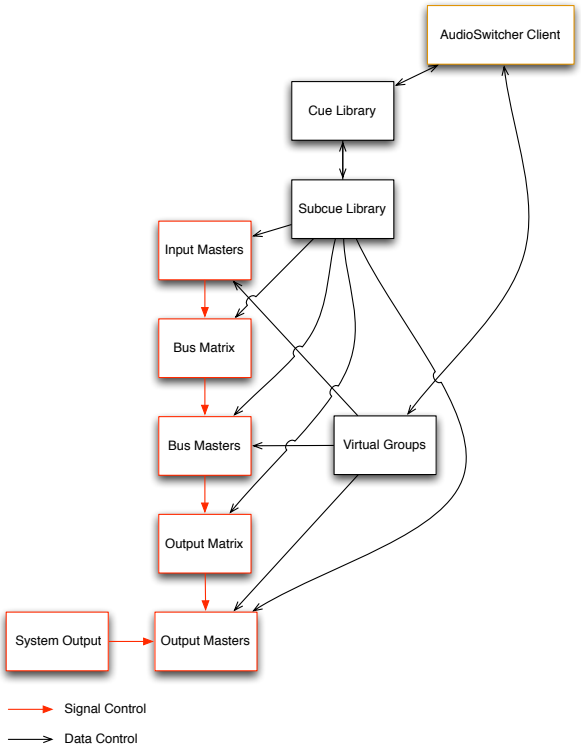


Figure 3: AudioSwitcher Server control and signal flow

3.2.1 Input/Bus/Output Masters, System Levels

Signal flow in AudioSwitcher Server resembles that of most DAW software tools, with input channels assigned to various bus channels, which are then sent on to output channels. Each input, bus, and output channel contains a post-fader signal-level meter, as well as controllable areas for its label, trim and fader levels, mute, solo, and delay states (Figure 4). Additionally, master control over all output channel level and mute states is provided in the System Output window. Level adjustments made to the system output are applied to all outputs at the pre-fader stage. All of the above controls can be manually adjusted or automated using AudioSwitcher Server's preset system (discussed in Section 3.3).

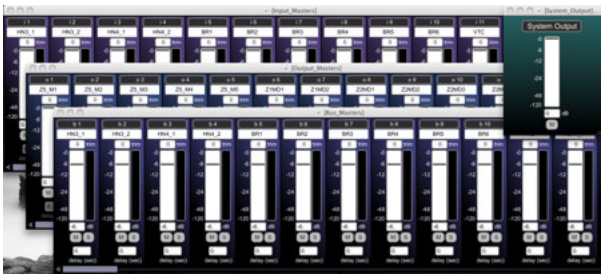


Figure 4: Input/Bus/Output Masters, and System Output

3.2.2 Bus and Output Matrices

The core of AudioSwitcher's signal flow centers around two variable-sized mixing matrices (Figure 5). By implementing mix matrices at two different stages in the signal flow, we provide a flexible vehicle for dealing with the complex mixing and routing scenarios encountered in multi-media spaces, effectively removing, for instance, the need for most output-to-input loopbacks (which the software also supports). All mix points in the matrices can be manually adjusted, or automated using the software's preset system (see Section 3.3).

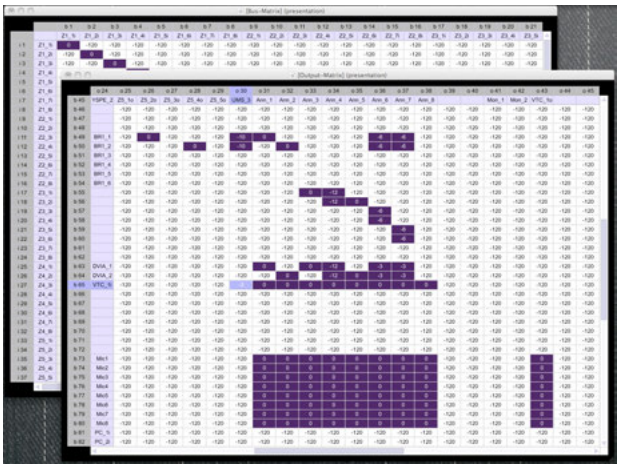


Figure 5: Bus Matrix and Output Matrix

3.3 Automation and Control

AudioSwitcher Server provides a control and automation layer for manual or remote event triggering and state changes over all facets of the software, such as Bus and Output Matrix assignments, Input/Bus/Output Masters labels, levels, mutes, solos, and delays, DAC on/off state, etc. Additionally, custom network messages can be created, stored and sent to remote destinations (triggering state changes in custom Max patches running within Asset Manager, for example). Once software and hardware configurations have been properly established, and signal control logic has been largely defined, three features within AudioSwitcher Server function as the primary vehicles for high level audio systems control: Cue Library, Subcue Library, and Virtual Groups.

3.3.1 Cue and Subcue Libraries, Virtual Groups

The Cue and Subcue Libraries provide display and control over user-defined presets. Cues exist solely as a means to store and recall one or more lower-level presets, called "subcues." Triggering a cue causes all subcues referenced by that cue to be sequentially recalled in a user-defined order (Figure 6). Whereas cues essentially act as subcue aggregators, subcues themselves apply automated control over virtually all aspects of AudioSwitcher Server's functionality; they do the actual work. Both cues and subcues can be manually triggered, or automated via calls from their control counterparts (i.e. cues referencing subcues, subcues referencing cues).

Virtual Groups provide high-level control over user-defined groups of input, bus, and output channels (Figure 6). In the Virtual Groups window, a user can manually set each group's label, trim, level, mute, and solo states, which in turn effect the corresponding states of all input/bus/output channels linked to that group. All virtual group controls can be manually adjusted, or automated via cues and subcues.

Establishing effective high-level control over complex audio systems is not unlike trying to hit a moving target, and as such it is important for a control system to support real-time user modification. Therefore, as mentioned earlier, cues, subcues, and virtual groups are all defined via text files that can be modified and reloaded while the server is running.

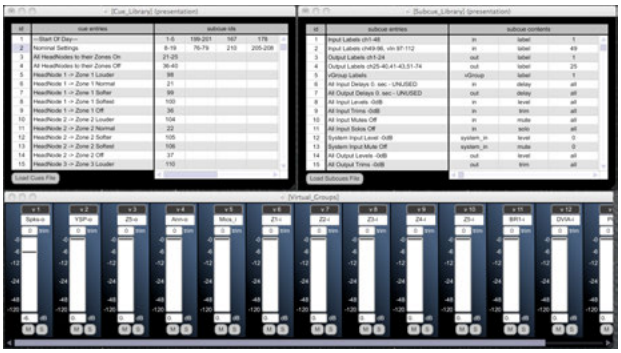


Figure 6: Cue Library, Subcue Library, and Virtual Groups

3.4 Network and MIDI i/o

In order to provide users the ability to customize the way in which they interface with AudioSwitcher Server, most of the software's control features can be configured to communicate with 3rd party hardware and software over network and MIDI protocols. Via the software's primary configuration file, a wide range of control points (such as virtual group faders, output master mutes, cues, subcues, etc.) can be "bound" to one or more user-defined network/MIDI senders and/or receivers, thus allowing a user to easily set up one and two-way real-time connections with external software and hardware, exposing as few or as many server control points as is appropriate for the situation. Using this functionality it is possible, for example, to create a simple control interface on the iPhone that remotely triggers cues and adjusts output levels, or to use faders on a MIDI controller to both display and control all virtual group fader and mute states. The methods by which network and MIDI "bindings" are established in the configuration file are documented for all relevant controls in the software, and should therefore provide a vehicle for the majority of custom user extensions to the control system.

4. ASSET MANAGER

In this section, we present the overall functionalities and the new workflow introduced by Asset Manager's framework.

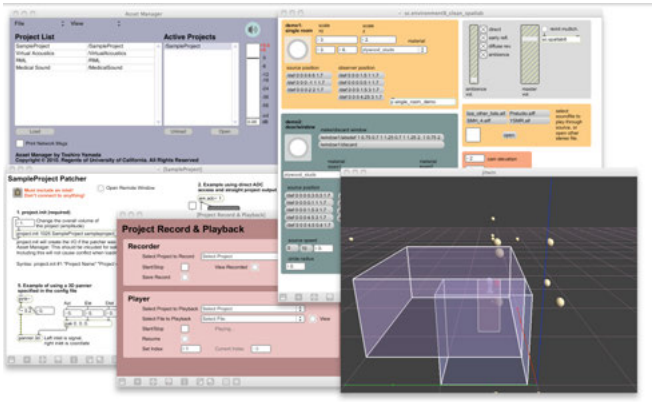


Figure 7: Asset Manager software with projects

4.1 Overview

Asset Manager is built to deal with multiple audio projects in an environment where projects need to be loaded on demand. We use Asset Manager extensively with virtual reality environments and other graphics engines, where each project requires a custom Max patch. Since each system has its own optimal spatialization setup, maintaining multiple versions of the same project implemented on different systems can quickly become cumbersome and time-exhaustive. Asset Manager addresses these difficulties by providing a framework that abstracts panning and signal flow, and helps optimize production workflow for complex sound systems.

Summary of Functionality:

- Configuration files for i/o setup, project definitions, signal paths and panning methods
- Versatile spatialization and i/o abstraction layers
- Network message specification in Open Sound Control protocol to control behavior of Asset Manager and projects
- Mixing control for each project and master outputs
- Recording and playback of network messages
- Built-in objects for spatialization signal processing, such as distance simulation, air absorption, Doppler effect, and source direction simulation

4.2 Signal Flow

4.2.1 Project to System Outputs

All audio signals from projects go through Asset Manager's system outputs layer, which serves as a final gain control stage before reaching the audio interface. Asset Manager provides a collection of Max patch abstractions that allow projects to utilize the software's signal paths. Furthermore, project volumes can be mixed independently from one another.

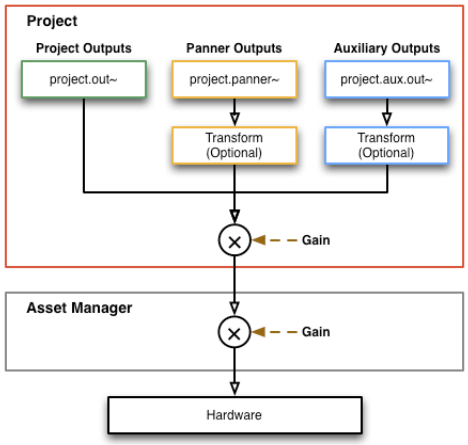


Figure 8: Project signal flow diagram

4.2.2 Project Signal Paths

Within a project, there are three paths a signal can take before reaching the system output. These paths bring logic separation and system abstraction that can be uniquely configured for different systems (Figure 8).

The main project outputs are accessed via the `project.out~` object. Signals passed to project outputs are routed directly to the main output without additional signal processing. In the project configuration file, channel IDs are mapped to hardware output channels. By using an indirect channel ID mapping (from project layer to hardware), the signal chain becomes independent from a specific audio system, and projects can be shared amongst different systems without having to modify the Max patch. This philosophy of

abstraction is used throughout Asset Manager's signal flow design. Main project outputs can be used to route static audio sources, such as voice-overs, which are commonly routed to a single loudspeaker (e.g. to the center channel in a 5.1 surround sound setup).

Panner outputs are used to spatialize the sound – or "pan" the sound – using the `project.panner~` object. The panning method implemented in a project can be anything from stereo, 5.1 surround sound, Ambisonics, HRTF binaural, to custom implementations; the object is abstract and has no implementation on its own. The implementation is defined in the project configuration file where other project settings are also configured. Additionally, an optional transform function can be added after the panner signal path. A transform function is a black box that includes any operation that processes the signal from inputs to outputs. For example, it can be a simple matrix that routes five input channels and six output channels, where the sixth channel has the sum of all inputs that is routed to the subwoofer.

Auxiliary outputs, accessed with the `project.aux.out~` object, are used when the main project outputs and panner outputs do not fulfill a particular need. `project.aux.out~` is used similar to `project.out~` but also includes an optional transformation found in `project.panner~`. For example, auxiliary outputs are useful when a project contains pre-panned sources, e.g. 5.1 surround sound tracks, which require a transform function to match source outputs to system outputs. If the target system is headphones, a transform may be a 5.1 surround sound to HRTF binaural encoder.

These three signal paths are simple, yet powerful enough to support a variety of output requirements. Using these abstract objects, projects can be ported to work in Asset Manager's framework and take advantage of its workflow.

4.3 Workflow for Complex Sound Server Requirements

4.3.1 Abstraction of Panning Method

Much of the strength of Asset Manager comes from the ability to isolate the implementation of the panning method and rapidly adapt new panning methods in real-time. This abstract layer has saved many hours reconfiguring new panners, keeping multiple copies of different versions, and trying out various methods that may or may not work in a given system. By specifying the panner in a plaintext file, version control and project sharing becomes much easier. Moreover, once a well-behaved panner is chosen for a system, new projects can easily take advantage of it without altering the original Max patch. Reusing well-tested panners can also reduce the likelihood of using them improperly, thus diminishing time spent debugging.

4.3.2 Network Communication

Asset Manager uses Open Sound Control extensively for network communications and can be used with various 3rd party hardware and software. Via OSC (over TCP/IP or UDP sockets), remote applications can control and automate core functionalities, such as (un)load projects, change master and project volumes, (un)mute projects, dis-/en-able signal processing, and more. Asset Manager also includes rich tools for working with OSC messages.

4.3.3 Record and Playback Network Messages

OSC messages can be recorded and played for each project, keeping the exact timing as the messages arrive. Playback is done on a loopback socket to simulate real network messages. This is useful for archiving important events, generating reference materials, and demonstrating and debugging projects. The last point has been especially useful at our laboratories, where we have various complex graphics display systems that use Asset Manager for audio contents management and synchronous audio playback. Asset Manager runs on a dedicated audio server and communicates with the visual systems remotely. Operating these systems involves complex steps with many potential points of failure, and debugging these problems can be a tedious and time-consuming process. Using the network record and playback feature, we can test Asset Manager projects independent from other components of the systems and determine the point of failure faster.

5. CONTROL PROXY

Control Proxy is a simple software utility designed to facilitate network communication between various software and hardware components within an audio system. This software fulfills three primary functions:

1. Acts as a hub for network traffic, redirecting incoming messages from a given source to one or more IP and port destinations.
2. Applies network protocol translation between incoming and outgoing messages.
3. Provides a console interface for manually sending messages to user-defined destinations using the appropriate protocols.

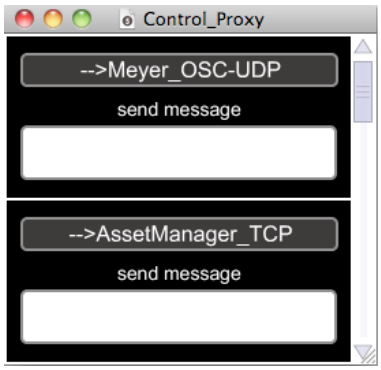


Figure 9: Control Proxy GUI

Logistically, this software provides developers with a *single* access point to a potential wide variety of destinations, without requiring them to know about destination-specific addresses, ports, and protocols. This allows audio staff to supply colleagues with a fixed set of ports at a single IP address that they can use when creating network links between the various non-audio software/hardware and audio systems. Text console network message windows serve as a convenient way to test interconnectivity, and to simulate commands coming from remote sources. Network protocol conversion serves to speed up development workflow when, for example, a software visualization tool sends messages only via UDP but needs to communicate with audio hardware that understands only TCP. All incoming to outgoing network redirection and conversion, as well labeling of each console window in the software's GUI, is defined in a simple plaintext configuration file similar in syntax to those used in AudioSwitcher Server and Asset Manager (Figure 9).

6. CONCLUSIONS

In this paper we have presented an open collection of software tools designed to enhance operations and development workflow on multi-functional audio systems within research facilities. The three software tools within our framework supply what we believe to be a core group of critical audio control capabilities – functionality required across a broad spectrum of audio systems scenarios, that if implemented well, have the potential for significantly streamlining audio systems operations and development pipelines. At the same time, the software remains sufficiently open and capable of supporting a wide variety of custom extensions.

In refining our notions of essential functionality, we have drawn upon our own experience operating on and developing for complex audio systems within dynamic and multi-media research environments. This functionality can be summarized as follows:

- Mixing/routing and system preset automation
- Dynamic audio asset management and control
- Network message routing/redirection and protocol conversion

Moving forward, we plan to improve upon a variety of features within the software, placing particular emphasis on ease of use. We will work towards a more complete integration of each software tools' configuration files into their respective GUI front-ends, allowing for easier real-time creation and editing without the need for script. We will also explore the benefits of replacing our own simple configuration file syntax with standardized file formats such as JSON, YAML, or XML.

Several functionality enhancements are planned for individual tools within the framework. In AudioSwitcher Server, we will implement "effects chain" functionality at the Input, Bus, and Output Masters stages, enabling users to dynamically load and modify a variable amount of custom Max patches or VST plug-ins at a particular stage in the signal flow. This will prove useful when, for example, you need to add a multi-tap delay or high-pass filter to a signal before sending it out to the loudspeakers. In Asset Manager, various spatial sound effects are in development, such as geometric acoustic simulations and multichannel reverberations. Furthermore, future releases will include a database backend for saving and restoring software states and accessing sound parameters and objects in real-time.

Finally, in an effort to improve documentation and discover overlooked core functionality, we hope to broaden the framework's user-base by releasing the tools open source to the community, and by continuing to work with research partners on implementations within their facilities. All software, as well as documentation, sample configuration files, and projects can be found at http://vis.kaust.edu.sa/tahakum.

7. ACKNOWLEDGMENTS

This project would not have been possible without the support of Steve Cutchin, Thomas A. DeFanti and all our colleagues at California Institute for Telecommunications and Information Technology and the KAUST Visualization Lab. Thanks to Paul Riker for his editorial assistance, and for helpful feedback on the software in its current state. Finally, we would like to thank Peter Otto of Calit2's Sonic Arts R&D group, for his essential insight and guidance throughout the design and implementation of the Tahakum framework.

8. REFERENCES

[1] S. Ellison, P. Otto, *Acoustics for reproducing sound at the visualization labs at the King Abdullah University of Science and Technology: A case study*. 159th Meeting of Acoustical Society of America: NOISE-CON 2010, Baltimore, USA, 2010 April 19-23.

[2] J. Fischer, F. Gropengiesser, S. Brix, *Cooperative Spatial Audio Authoring: Systems Approach and Analysis of Use Cases*. 126th AES Convention, Munich, Germany, 2009 May 7-10.

[3] F. Gropengiesser, K. Sattler, *An Extended Co-operative Transaction Model for XML*, Work-shop for Ph.D. Students in Information and Knowledge Management (PIKM'08), Napa Valley, USA, 2008 October 26–30.

[4] N. Humon et al. *Sound Traffic Control: An Interactive 3-D Audio System for Live Musical Performance*. Proceedings of the 1998 Conference on Auditory Displays, Glasgow, UK, 1998 November 1-4.

[5] F. Melchior, C. Sladeczek, A. Partzsch, S. Brix, *Design and Implementation of an Interactive Room Simulation for Wave Field Synthesis*. Proceedings of the AES 40th International Conference, Tokyo, Japan, 2010 October 8-10.

[6] D. Murphy and F. Rumsey, *A Scalable Spatial Sound Rendering System*. 110th AES Convention, Amsterdam, The Netherlands, 2001 May 12-15.

[7] T. Musil et al. *The CUBEmixer a performance, mixing and mastering tool.* Proceedings of the 2008 Linux Audio Conference, Cologne, Germany, 2008 Feb 28 - March 2.

[8] N. Peters et al. *A stratified approach for sound spatialization*. Proceedings of the 6th Sound and Music Computing Conference, Porto, Portugal, 2009 July 23-25.

[9] S. Wilson, J. Harrison. *Rethinking the BEAST: Recent developments in multichannel composition at Birmingham ElectroAcoustic Sound Theatre.* Organized Sound (2010) vol. 15 (03) pp. 239-250.

A Framework for Coordination and Synchronization of Media

Dawen Liang
Carnegie Mellon University
School of Music
5000 Forbes Ave, Pittsburgh, PA
dawenl@andrew.cmu.edu

Guangyu Xia
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave, Pittsburgh, PA
gxia@cs.cmu.edu

Roger B. Dannenberg
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave, Pittsburgh, PA
rbd@cs.cmu.edu

ABSTRACT

Computer music systems that coordinate or interact with human musicians exist in many forms. Often, coordination is at the level of gestures and phrases without synchronization at the beat level (or perhaps the notion of "beat" does not even exist). In music with beats, fine-grain synchronization can be achieved by having humans adapt to the computer (e.g. following a click track), or by computer accompaniment in which the computer follows a predetermined score. We consider an alternative scenario in which improvisation prevents traditional score following, but where synchronization is achieved at the level of beats, measures, and cues. To explore this new type of human-computer interaction, we have created new software abstractions for synchronization and coordination of music and interfaces in different modalities. We describe these new software structures, present examples, and introduce the idea of music notation as an interactive musical interface rather than a static document.

Keywords

Real-time, Interactive, Music Display, Popular Music, Automatic Accompaniment, Synchronization

1. INTRODUCTION

Computer music systems have been used extensively in interactive performances of cutting-edge electro-acoustic music, and also in some advanced systems that model the traditional role of the accompanist in Western art (or "classical") music [13]. In the realm of popular music, computers have had their largest impact through new instruments (almost every electronic instrument now has some sort of embedded computer). The concept of "instrument" has been extended to include the laptop computer, especially in loop-based music related to the DJ phenomenon. We believe that there are untapped possibilities in more traditional popular music forms such as rock, jazz, and folk music. There are opportunities here for innovative applications of highly intelligent and coordinated computer music systems [11]. In both rehearsal and live performance, computers could contribute to make new sounds possible, fill in for missing musicians, and ultimately to inspire new musical directions based on new capabilities and concepts from new technologies.

To bring computers into the realm of popular music performance, certain problems must be addressed. The main problem is that popular music timing is organized around a tight synchronization to beats. When live musicians are involved, the tempo is not perfectly steady, and humans have a difficult time synchronizing to an unyielding computer time-keeper. At the same time, computers cannot reliably adapt to human tempo variations. Another significant problem is the improvisation and decision-making that goes on in many live performances. It would be simple to prepare computers with fixed sequences, but what happens when the vocalist comes in a measure late or the bandleader signals to play another chorus? These problems are even more difficult given the amount of structure in popular music. Musicians and their audience know when performers are tightly synchronized in terms of rhythm and harmony. We cannot expect computers to improvise freely or "play by ear." Instead, they must understand, communicate, and synchronize at the level of beats, measures, and pre-determined musical structure such as sections and chord progressions.

Imagine a popular music performance system that could play different representations of music including MIDI, audio, guitar tabs, etc. as accompaniment and quickly adjust its tempo to follow the performer. Furthermore, the performance system could display an image of the score and automatically turn pages. In rehearsals, the computer could cover missing parts, especially for individual practice, and in live performance the computer could play additional parts not covered by human performers. The computer could be directed in part by pointing to locations in the score image, and the computer could confirm its location or intention to play by highlighting locations in the score.

To create such a system, we must coordinate time among different media. We would like to do this systematically and modularly so that new media can be added to the system without rewriting all the low-level, time-critical software. For example, one might want to synchronize video or lyrics to live music. How would this fit into an audio framework? This paper presents a flexible, beat-based "virtual time" framework to meet this challenge. One of the interesting aspects of this work is the two-way coordination of a visual score with a live computer performance, creating an interesting human-computer interface. Using a music notation display, the human can direct the performance to a location in the score, and the computer can give feedback to the human as to the current score location.

The next section presents related work. Section 3 describes how synchronization is achieved by scheduling computation according to piece-wise linear maps between time and beat

NIME'11, 30 May–1 June 2011, Oslo, Norway.

position. Techniques to keep these maps smooth in the face of latency and changing tempo estimates are presented. Section 4 describes a modular software framework for controlling multiple synchronized media player objects. Section 5 discusses the use of music notation as a bi-directional graphical interface for controlling music performance and monitoring the status of a computer performer. Our current implementation is discussed in Section 6. Finally, conclusions are presented in Section 7.

2. RELATED WORK

Much work has been done in the area of music performance systems. For example, automatic accompaniment systems for classical music performance [8], [9], [17], [18] and real-time music composition and performance systems [20] have been used and studied for many years. Related work exists in the area of music conducting. The work by Lee, Karrer, and Borchers [16] is especially relevant to our work in its discussion of synchronization of beats and smooth time map adjustment, and recent work [3], [14] discusses both tempo adjustment and synchronized score display, using an architecture similar to ours. However, the particular problems of popular music seem largely to be ignored. Of course, one simple way to incorporate computers in live popular music performance is to change the problem: humans can adapt to the steady time of the computer by listening to drums or a click track, and a fixed structure enables computers to play fixed sequences. Ableton Live [1] is an example of software that uses a beat, measure, and section framework to synchronize music in live performance, but the program is not well-suited to adapting to the tempo of live musicians. Robertson and Plumbley used a real-time beat tracker in conjunction with Ableton Live software to synchronize pre-recorded music to a live drummer [19]. Our goal is to create a more autonomous "artificial performer" that does not require a human operator sitting at a computer console, but rather uses more natural interfaces for direct control and more sophisticated listening and sensing for indirect control.

3. MEDIA SYNCHRONIZATION

The main role of our architecture is to synchronize media in multiple modalities. Because we assume popular music forms, we also assume a common structure of beats and measures across all media. Thus time is measured in beats. The basis for synchronization is a shared notion of the current beat and the current tempo. Beats are represented by a floating point number, hence they are continuous rather than integers or messages such as in MIDI clock messages. Also, rather than update the beat number at frequent intervals, we use a continuous linear mapping from time to beat. This mapping is conveniently expressed using three parameters (b_0, t_0, s):

$$b = b_0 + (t - t_0) \times s \quad (1)$$

where tempo s is expressed in beats per second, at some time in the past beat b_0 occurred at time t_0, the current time is t, and the current beat is b. (One could also solve for b_0 when $t_0 = 0$ to eliminate one parameter, but we find this formulation more convenient.

One advantage of this approach is that it is almost independent of latency. One can send (t_0, b_0, s) to another computer or process and the mapping will remain valid regardless of the transmission latency. There is an underlying assumption of a shared global clock (t), but accurate clock synchronization is straightforward [5] and can be achieved independently of media synchronization, thus making the system more modular. When parameters change, there can be a momentary disagreement in the current time among various processes, but this should be small given that tempo is normally steady. We will see below how these slight asynchronies can be smoothed and do not lead to long-term drift.

In our system, media players schedule computation to affect the output at specific beat times. For example, an audio player may begin a sample playback at beat 3, or a MIDI player may send a note-on message at beat 5. The current beat time b in Eq. 1 refers to the beat position of media which are being output currently, e.g. the beat position corresponding to the current output of a digital-to-analog converter (DAC). Time-dependent computation of media must of course occur earlier. For example, if the audio output buffer contains $0.01s$ of audio, then computation associated with beat b should be performed $0.01s$ earlier than b. Thus, given a player-specific latency l, we need to compute the real time t at which to schedule a computation associated with beat b. The following formula is easily derived:

$$t = t_0 + (b - b_0) / s - l \quad (2)$$

We simply map the beat position b according to (b_0, t_0, s), and then subtract the latency l to get the computation time t.

3.1 Estimating the Mapping

Our current system relies on a simple foot pedal to tap beats. A linear regression over recent taps is used to estimate the mapping from beat to time (*i.e.* to estimate t_0, b_0, and s). At this stage, successive beats are numbered with successive integers, but these start at an arbitrary number. Once the tempo and beat phase is established, there must be some way to determine an offset from the arbitrary beat number to the beat number in the score. This might be determined by a cue that tells when the system should begin to play. In other cases, especially with a foot-pedal interface, the system can be constructed to, say, start on the third foot tap.

We believe that audio analysis could be used to automate beat identification to a large extent, and we are investigating combinations of automated and manual techniques to achieve the high reliability necessary for live performance. The important point here is that *some* mechanism estimates a local mapping between time and beat position, and this mapping is updated as the performance progresses.

3.2 Tempo and Scheduling

Schedulers in computer music systems accept requests to perform specific computations at specific times in the future. Sometimes, the specified time can be a "virtual" time in units such as beats that are translated to real time according to a (possibly varying) tempo, as in Eq. 2. Previous architectures for handling tempo control and scheduling [2] have assumed a fixed and uniform latency for all processing. Under this assumption, there are some interesting fast algorithms for scheduling [9]. An important idea is that all pending events (callbacks) can be sorted according to beat time and then one need only worry about the earliest event. If the tempo changes, only the time of this earliest event needs to be recomputed. Unfortunately, when event times are computed according to Eq. 2, the earliest pending event can change when tempo changes. Therefore, we need to rethink scheduling structures of previous systems. The non-uniformity of latency is a real issue in our experience because audio time-stretching can have a substantial latency due to pre-determined overlap-add window sizes, page turning might need to begin seconds ahead of the time of the first beat on the new page, etc.

A second problem is that when the time-to-beat mapping is calculated from linear regression, there can be discontinuities in the time-to-beat-position function that cause the beat

position to jump forward or backward instantaneously. Most media players will need to construct a smooth and continuous curve that approximates the estimated time-to-beat mapping. We do this using a piece-wise linear time-to-beat map, adjusting the slope occasionally so that the map converges to the most recent linear regression estimate of the mapping.

Figure 1 illustrates this process. The lower line represents an initial mapping according to Eq. 1. Imagine that at time t_1, a new beat has resulted in a new linear regression and a new estimate of the time-to-beat map shown in the upper line. This line is specified by an origin at (t_e, b_e) and a slope (tempo) of s_e beats per second. The problem is that switching instantly to the new map could cause a sudden jump in beat position. Instead of an instant switch, we want to "bend" our map in the direction of the new estimate. We cannot change the current (lower) map immediately at t_1 because output has already been computed until t_1+l, where l is the latency. For example, if audio output has a 0.1s latency, then samples computed for beat position b at time t_1 will emerge at t_1+0.1. Thus, the earliest we can adjust the map will be at time t_1+l corresponding to beat b. Let us call the new map parameters t_n, b_n, and s_n. Since the current map passes through (t_1+l, b), we will choose this point as the origin for the new map (Eqs. 3, 4, 5) leaving only s_n to be determined.

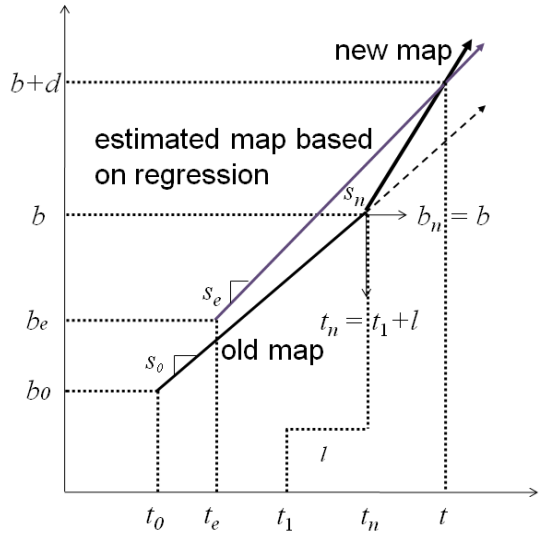


Figure 1. Modifying the local time-to-beat mapping upon receipt of a new regression-based mapping estimate.

$$b = b_0 + (t_1 + l - t_0) \times s_0 \quad (3)$$
$$t_n = t_1 + l \quad (4)$$
$$b_n = b \quad (5)$$

We choose s_n so that the new time map will meet the estimated (upper) time map after d beats, where larger values of d give greater smoothing, and shorter values of d give more rapid convergence to the estimated time map. (We use 4 beats.) In practice, we expect a new linear regression every 2 beats (cut time), thus the new time map will only converge about half way to the estimated map before this whole process is repeated to again estimate a new map that "bends" toward the most recent time-to-beat map estimate.

To solve for s_n, notice that we want both the upper regression line and the new time map to meet at (t, b_n+d), so we can substitute into Eq. 1 to obtain an equation for each line. This gives two equations (Eqs. 6, 7) in two unknowns (t and s_n):

$$b_n + d = b_e + (t - t_e) \times s_e \quad (6)$$
$$b_n + d = b_n + (t - t_n) \times s_n \quad (7)$$

Solving for s_n gives us Eq. 8:

$$s_n = \frac{d}{t_e s_e - t_n s_e - b_e + b_n + d} s_e \quad (8)$$

Under this scheme, we set (b_0, t_0, s_0) to (b_n, t_n, s_n) after each new estimated time map is received. Because of Eq. 3, these parameters depend on latency l, which can differ according to different players. It follows that different media will follow slightly different mappings. This can be avoided, and things can be simplified by giving all media the same latency. For example, MIDI messages can be delayed to match a possibly higher audio latency. In any case, time map calculation is still needed to avoid discontinuities that arise as new beat times suddenly change the linear regression, so we prefer to do the scheduling on a per-player basis, allowing each player to specify a media-dependent latency l. Note that (b_n, t_n, s_n) describes the *output* time for media. Given latency l, computation must be scheduled early according to Eq. 2. Equivalently, we can shift the time map left by l.

4. MODULAR STRUCTURE

Our system is organized as a set of "Player" objects that interact with a "Conductor" object that controls the players. The Conductor provides a central point for system control. The Players also use a real-time scheduler object to schedule computation according to Eq. 2. The interface and interaction between the Conductor and Players is illustrated in Figure 2.

4.1 The Player Class

A Player is any object such as an audio or MIDI sequencer that generates output according to the current tempo and beat position. A Player can also generate visual output, including page turning for music notation or an animated display of the beat.

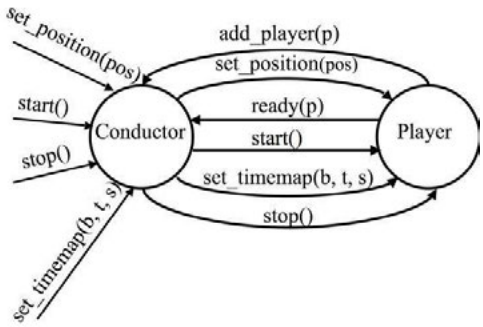


Figure 2. Interfaces for Conductor and Player objects.

Every "player" implements four methods used for external control: *set_position*(*pos*), *start*(), *stop*(), and *set_timemap*(*b*, *t*, *s*). The *set_position*(*pos*) method is a command to prepare to output media beginning at beat position *pos*. This may require the player to pre-load data or to output certain data such as MIDI controller messages or a page of music notation. The s*tart*() method is a command to begin output according to the current tempo and the mapping from time to beat position. The playback can be stopped with the *stop*() command. Note that stopping (sound will cease, displays indicate performance has finished) is different from setting the tempo to zero (sound sustains, displays are still active), so we need explicit start and stop signaling. The *set_timemap*(*b, t, s*) method updates the mapping from real time to beat position to the linear function that passes through beat b at time t with slope s (in beats per second). This is how the new linear regression data (t_e, b_e, s_e) described in the previous section is transmitted to each Player.

Note that the external interface to Players concerns time, beats, and control, but says nothing about media details. In this

way, new players can be added in a modular fashion, and the details of player operation can be abstracted from the overall system control.

4.2 The Conductor Class

The role of a Conductor is to provide a single point of control and synchronization for all players. The Conductor methods include the same *set_position*(*pos*), *start*(), *stop*(), and *set_timemap*(*b, t, s*) methods as do Player objects. These methods are to be used by higher level control objects. For example, a graphical user interface may have a conventional play/stop/pause/rewind interface that is implemented by Conductor methods. Alternatively, a more intelligent system might use automatic music listening, gestures, or other ways to determine when and where to start and stop. In addition, an *add_player*(*p*) method allows new Player objects to add themselves to the list of Players managed by a single Conductor.

4.3 Scheduling

We assume the existence of a real-time scheduler object [9] to be used by Players. A typical player has computation to perform at specific beat times. Usually, a computation will perform some action needed at the present time, followed by the *scheduling* of the *next* action. The scheduler's role is to keep track of all pending actions and to invoke them at the proper time, thus eliminating the need for Players to busy wait, poll, or otherwise waste computer cycles to ensure that their next computation is performed on time. Players use Eq. 2 to determine the real time t at which to perform an action scheduled for beat position b.

4.4 Coordination of Media

An important feature of the framework is that it coordinates media of different forms – midi, audio, score, etc. – in real-time performance. In this section, we will discuss the details of time synchronization.

4.4.1 Shared Time System

As introduced in Section 2, the framework is based on a shared notion of beat position, i.e. all the players controlled by the Conductor share the same beat position. The beat information for most MIDI is easy to extract because it is normally encoded in a Standard MIDI File. Audio and score images are more problematic.

For audio, we must have auxiliary information that encodes a mapping from beat position to audio time. An audio Player can then use time-stretching algorithms to adjust the audio speed to synchronize to a live performance. The audio can be recorded at constant tempo, e.g. using a click track, so that beat positions can be calculated directly from the known tempo. Alternatively, audio can also be labeled by manual tapping, by automatic beat tracking (for music where this is possible), or by automatic alignment [13] to audio or MIDI for which beat times are known.

For music notation, we can use structured documents such as MusicXML [6] or unstructured scanned images. In principle, structured score documents have all the information needed to map from beats to page numbers and positions, but in practice, rendering music notation is difficult and there is no readily available software that can be adapted to our purpose. Instead, we let users indicate the time signature (or by default 4/4) and manually label the start position of each measure to construct a mapping from beats to image position. Using this information, the score Player can convert beat positions to approximate page locations. In the future, we hope to adapt some optical music recognition (OMR) software to detect systems and bar lines to speed up the process of annotating score images. OMR combined with symbolic music to audio alignment is another promising approach to label scanned music notation [15].

4.4.2 Distributed Computation

The framework supports distributed computation or computation in separate threads on multi-core computers. Coordination and synchronization is often difficult in distributed systems because of unknown communication latency. In our approach, communication latency is not critical. Communication latency certainly affects the responsiveness of the system, but unless tempo changes drastically, beat positions are predictable in the near future. Instead of transmitting beat times, we transmit *mappings* from global time to beat position. These mappings are expressed with respect to a shared global clock, and they do not change even if their delivery is delayed. Any two processes that agree in terms of their real clock time and their mapping (t_0, b_0, s) will agree on the current beat position.

In a distributed implementation, the Conductor communicates via (reliable) messages with Players, and Players rely on local schedulers to activate timed computations (see Figure 3). If the schedulers are on separate computers, the computer real-time clocks must use a clock synchronization protocol to ensure that every scheduler agrees on the real clock time.

We have found it easy to synchronize clocks at the application level. For example, designated *slave* machines send a request to a *master* for the time, and the master time is returned. This round trip time is usually less than a few milliseconds, and the *slave* can set its clock assuming a communication latency of half the round trip time. This can easily produce synchronization to within 1ms. If the round trip time is longer than normal, the slave simply assumes that an unexpected network delay has made the result unreliable, ignores the result, and tries again. More elaborate techniques based on averaging and estimating clock drift can even synchronize clocks to microseconds if needed [5].

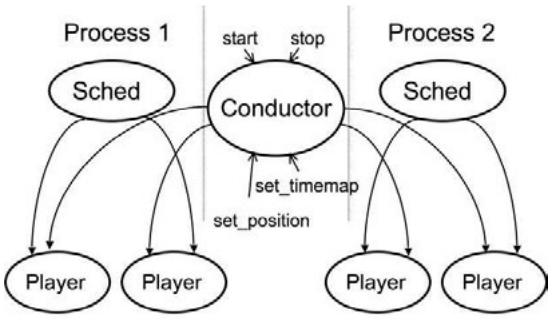


Figure 3. Distributed message-based implementation.

4.4.3 Static vs. Dynamic Scores

Even after providing mappings from beat position to specific time and spatial coordinates of different media, there is an important difference between scores and most other media that we must deal with. Scores are a bit like programs that must be "executed" to determine a music performance. Repeats and the "dal segno al coda" are forms of looping behavior. First and second endings and the coda are forms of conditional behavior based on the loop count. Thus, the score is a "static" representation of music in the sense of static code, and an audio file or MIDI file is a "dynamic" representation of music in the sense of dynamic or run-time program behavior. Because of repetition, there is a one-to-many association between static score position and dynamic beat position. Our current system implements conventional music control

structures, but real scores often resort to informal instructions that are difficult to formalize.

5. NOTATION AS INTERFACE

The idea of electronic display of music is not a new idea [4], [7] [15] [17], but we introduce the notion of active music notation as a bi-directional human-computer interface.

5.1 Location Feedback and Page Turning

In an interactive music system where synchronization is key, it is important for performers to communicate their coordination with the group. For example, when it is time for a guitar solo, the vocalist and guitarist might look at each other to acknowledge that both musicians expect the solo. If the vocalist's gestures instead indicate he or she will sing another chorus, the guitarist might hold off until later. In a similar way, it is important for the computer to signal its current position to human players so that they can either adapt to the computer or provide some override to steer the computer back into synchronization.

Music notation provides an attractive basis for communication because it provides an intuitive and human-readable representation of musical time, it is visual so that it does not interfere with music audio, and it provides both history and look-ahead that facilitates planning and synchronization. Given a mapping from beat position to score image location, it is easy to display the real-time beat position directly on an image of the score. Human musicians can then notice when the measure they are reading does not correspond to the measure that is highlighted and take corrective action.

Another possibility is automatic page turning, which was introduced in early computer accompaniment systems. For example, SmartMusic [17] uses the Finale notation engine to show scores and score position in real time as it follows a soloist in the score. In our framework, page turning is easily controlled by the Conductor. Just like scheduling an event from the MIDI player, the score player can also schedule a "scrolling-up" event.

Various schemes have been implemented for "page turning" on a display screen of limited size. It is well known that musicians read ahead, so it is essential to display the current music as well as several measures in the future. The most common approach is to split the screen into top and bottom halves. While the musician reads one half, the computer updates the other half to the next system(s) of music. Other solutions include: scrolling up at a constant speed, scrolling up by one system when it is finished, scrolling at a variable speed which is proportional to the tempo, and scrolling an "infinitely wide" score horizontally. Our implementation presents multiple "slices" of the score on the screen (see Figure 4), but we plan to experiment with different approaches.

5.2 Selecting Locations from Notation

In addition to affording computer-to-human feedback, music notation can be used as an "input device," for example to indicate where to begin in a rehearsal. Our system has start positions for every measure stored as coordinates (page, x, y). When we point to the position where we would like to start, the system can map the position to a beat number and use the Conductor's *set_position* method to prepare all Players to start from that location. This will also indicate the position in the score, giving a confirmation to the user that the correct location was detected.

6. IMPLEMENTATION

We have implemented a prototype system in Serpent [10], a real-time programming language inspired by Python. Our system follows the architecture described earlier, with classes *Conductor*, *Player*, and *Time_map*. The *Player* class is subclassed to form *Midi_player*, *Score_player* (a music notation display program), and *Posn_player* (to display the current position). Each player implements methods for *set_position*, *start*, *stop*, and they all inherit a method for *set_timemap* that adjusts each local player time map to converge to that of the conductor.

The score player class is the most complex (about 2400 lines of Serpent code). It displays music notation, turning "pages" automatically according to score position given by the conductor. The music notation comes from image files (e.g. jpeg or png), which are manually annotated. The score player includes graphical annotation tools to: (1) indicate the staff height, (2) subdivide the score into systems, (3) mark bar lines, (4) mark repeat signs, endings, *D.S.*, *coda*, and *fine*, (5) mark a starting measure, and (6) add arbitrary free hand and text annotations. (See Figure 4.)

After annotating the score, the score player sorts measures, repeats, and other symbols to form a representation of the *static* score. It can then compute a *dynamic* score by "unfolding" the repeats and computing a list of dynamic measures. The score player also scales the music notation images to fit the width of the display and divides the images into slices that are stacked vertically on the display.

There are many possibilities for music scrolling and page-turning. In the current implementation, we divide the screen into thirds and always display the previous, current, and next sub-pages. For example, the initial display shows the first 3 sub-pages, in the order 1-2-3. When the player advances to the third sub-page, the display is updated to show 4-2-3. The player continues reading sub-page 4 at the top of the display, at which time the display updates to 4-5-3, etc.

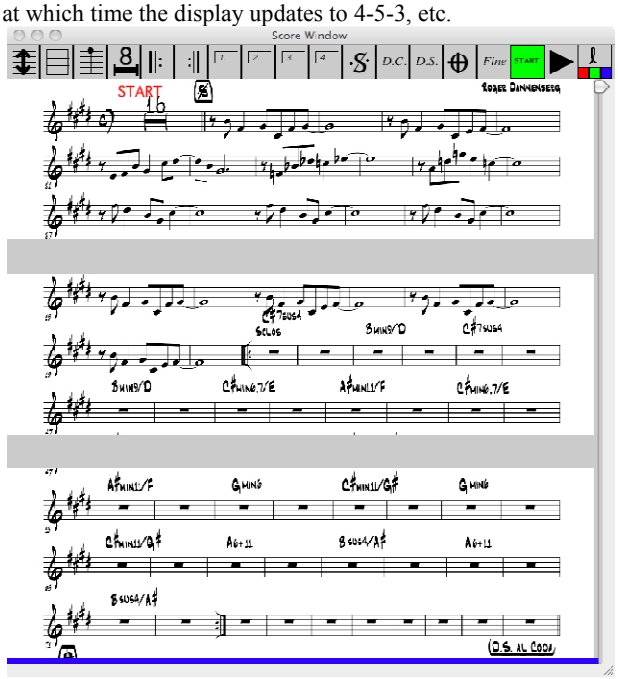


Figure 4. Score display showing editing toolbar at top and a vertical division into thirds.

We have also implemented a player for multi-channel audio that applies high-quality time stretching to each channel according to a time map [12]. However, this system is not yet integrated into the conductor/player framework. Finally, we have implemented a tempo control object that accepts beats from a space-bar or foot pedal, rejects outliers, and performs linear regression on recent taps to estimate a time map.

7. CONCLUSIONS AND FUTURE WORK

In conclusion, our framework implements a beat-based strategy for coordination and synchronization of media in real-time performance. We convert the music notation from images of score to a dynamic interactive display medium interface, which could cooperate with other forms of media to create a human-computer music performance system.

In the future, we plan to focus on applications that integrate music notation with audio playback, gaining practical experience using music notation as an interactive medium. We plan to test and evaluate different methods of music scrolling and assess whether music notation on a touch display can be used to control a computer musician during real performances.

Considering the complexity for prototyping and testing, our most recent system is based on MIDI and score images. As the framework structure becomes mature, we will integrate audio playback using PSOLA [21] and phase vocoder [16] time-stretching techniques. We will also need to implement editing techniques to label audio.

The scheduling and music notation framework described here is part of a larger overall architecture that facilitates music representation, preparation, and performance, and there are many more components required to achieve the kind of music production and performance flexibility that we envision. However, even based on this framework, many applications can be developed. For example, we could automatically align rehearsal recordings to MIDI files to quickly (and roughly) label audio. Then, we could listen to particular parts by pointing to the score, comparing the rehearsal performance of the same piece from two different days, etc. The flexibility of the framework provides many possibilities for future work.

8. ACKNOWLEDGMENTS

Thanks to Ryan Calorus, who implemented our first experimental music display, and Nicolas Gold for valuable discussions. Our first performance system and the music display work were supported by Microsoft Research and the Carnegie Mellon School of Music. Zplane kindly contributed their high-quality audio time-stretching library for our use. Current work is supported by the National Science Foundation under Grant No. 0855958.

9. REFERENCES

[1] Ableton. *Ableton reference manual (version 8).* http://www.ableton.com/pages/downloads/manuals (2011).

[2] Anderson, D. and Kuivila, R. A system for computer music performance. *ACM Transactions on Computer Systems*, Volume 8 Issue 1 (1990), pp. 56-82.

[3] Baba, T., Hashida, M., and Katayose, H. "VirtualPhilharmony": A Conducting System with Heuristics of Conducting an Orchestra. *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010),* ACM Press, 2010, *263-270.*

[4] Bainbridge, D. and Bell, T. An ajax-based digital music stand for greenstone. *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries* (JCDL '09), ACM, New York (2009), pp. 463-464.

[5] Brandt, E. and Dannenberg, R. Time in distributed real-time systems. *Proceedings of the 1999 International Computer Music Conference*, ICMA, San Francisco (1999), pp. 523-526.

[6] Castan, G., Good, M. and Roland, P. Extensible markup language (XML) for music applications: An introduction. *The Virtual Score*, MIT Press, Cambridge, MA, 2001, pp. 95-102.

[7] Connick, H. Jr. System and method for coordinating music display among players in an orchestra. US Patent #6348648 (2002).

[8] Cont, A. ANTESCOFO: Anticipatory synchronization and control of interactive parameters in computer music. *Proceedings of International Computer Music Conference* (*ICMC*), ICMA, San Francisco, 2008.

[9] Dannenberg, R. Real-time scheduling and computer accompaniment. In *Current Directions in Computer Music Research*, edited by Max. V. Mathews & John R. Pierce, MIT Press, Cambridge, MA, 1989, pp.225-261.

[10] Dannenberg, R. A Language for Interactive Audio Applications. *Proceedings of the 2002 International Computer Music Conference*, ICMA, San Francisco, 2002, 509-515.

[11] Dannenberg, R. New interfaces for popular music performance. *Seventh International Conference on New Interfaces for Musical Expression: NIME 2007*, New York, NY, 2007, 130-135.

[12] Dannenberg, R. A Virtual Orchestra for Human Computer Music Performance. *Proceedings of the 2011 International Computer Music Conference,* (to appear).

[13] Dannenberg, R. and Raphael, C. Music score alignment and computer accompaniment. *Commun. ACM* 49, 8 (August 2006), pp. 38-43.

[14] Katayose, H. and Okudaira, K. Using an Expressive Performance Template in a Music Conducting Interface. *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04),* (Hamamatsu), ACM Press., 2004, 124-129.

[15] Kurth, F., Müller, M., Fremerey, C., Chang, Y. and Clausen, M. Automated synchronization of scanned sheet music with audio recordings. *Proceedings of ISMIR*, Vienna (2007), pp. 261-266.

[16] Lee, E., Karrer, T. and Borchers, J. Toward a framework for interactive systems to conduct digital audio and video streams. *Computer Music Journal*, 30(1) (Spring 2006), pp. 21-36.

[17] MakeMusic, Inc. *SmartMusic interactive music software transforms the way students practice* (web page), http://www.smartmusic.com (2011).

[18] Raphael, C. Music Plus One: A system for flexible and expressive musical accompaniment. *Proceedings of the International Computer Music Conference*, (Havana, Cuba), ICMA, San Francisco, 2001.

[19] Robertson, A. and Plumbley, M. D. B-Keeper: A beat tracker for real time synchronisation within performance. *Proceedings of New Interfaces for Musical Expression* (*NIME 2007*), New York, NY, USA, (2007), pp 234-237.

[20] Rowe, R. *Interactive Music Systems*. MIT Press, Cambridge, MA (1993).

[21] Schnell, N., Peeters, G., Lemouton, S., Manoury, P., Rodet, X. Synthesizing a choir in real-time using Pitch Synchronous Overlap Add (PSOLA). *International Computer Music Conference (ICMC)*, (Berlin), ICMA, San Francisco, 2000.

Satellite CCRMA: A Musical Interaction and Sound Synthesis Platform

Edgar Berdahl
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
Stanford, CA, USA
eberdahl@ccrma.stanford.edu

Wendy Ju
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
Stanford, CA, USA
wendyju@ccrma.stanford.edu

ABSTRACT

This paper describes a new Beagle Board-based platform for teaching and practicing interaction design for musical applications. The migration from desktop and laptop computer-based sound synthesis to a compact and integrated control, computation and sound generation platform has enormous potential to widen the range of computer music instruments and installations that can be designed, and improves the portability, autonomy, extensibility and longevity of designed systems. We describe the technical features of the Satellite CCRMA platform and contrast it with personal computer-based systems used in the past as well as emerging smart phone-based platforms. The advantages and trade-offs of the new platform are considered, and some project work is described.

Keywords

NIME, Microcontrollers, Music Controllers, Pedagogy, Texas Instruments OMAP, Beagle Board, Arduino, PD, Linux, open-source

1. INTRODUCTION

As instructors of CCRMA's course in Physical Interaction Design for Music [17], we have noticed that many of incredibly novel and innovative musical instruments and installations created in our course over the years have very short lives; even projects demonstrated at the NIME conference [18, 15, 19, 8, 3, 1, 4, 14, 6] are often inoperable just a year afterwards. Conversations with researchers and musicians at a wide variety of other institutions indicate that this is phenomenom is not isolated to CCRMA alone, and that, even for the very-motivated, it takes enormous effort to keep NIME instruments in a functional and performable state. This lack of longevity and robustness creates a situation where 1) the quality of music produced by NIME instruments is limited by the fact that the instruments do not last long enough for musicians to develop expertise, to compose great scores, or refine the initial instrument designs, and 2) the quality of shared learning from NIME instruments is limited by the fact that the instruments are seldom transfered or built upon except by the same researchers who originated a particular design. While the rapid obsolescence of new musical controllers might be caused by the relatively short attention span of students or by the focus on novelty within the community, we believe that some element of the blame might be placed on the computer platforms that are at the heart of most NIME instruments.

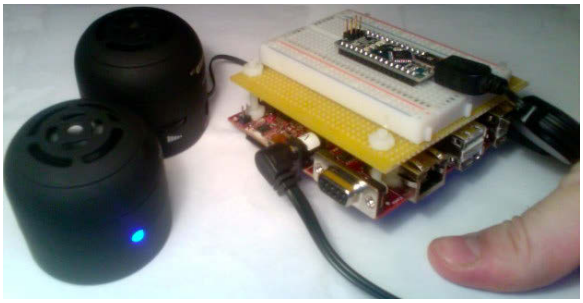


Figure 1: Satellite CCRMA

Satellite CCRMA (Figure 1) is a musical interaction design platform designed to support the creation of new instruments for musical expression as well as sound installations. It incorporates a single-board OMAP-based Linux computer, an Arduino-based microcontroller, and a breadboard for electronics prototyping. By creating a platform which includes a small, inexpensive and autonomous Linux computer, we hope to free NIME designs from the constraints and obstacles associated with platforms which use more general-purpose laptops and desktop workstations.

This paper describes the Satellite CCRMA platform design, its underlying rationale, our initial forays into introducing this platform to students in our Physical Interaction Design for Music course, and what we have learned so far. While it will be many years before we can empirically determine whether our platform is indeed longer-lived than the alternatives, our interim assessments of the affordability, adoptability and the extensibility of the platform give us great optimism that that the platform will be of great use to a large number of people within the NIME community and beyond.

2. PLATFORM DESCRIPTION

2.1 Hardware

The Satellite CCRMA platform is centered around the OMAP35x embedded processor line from Texas Instruments, which can run Linux. We currently use the Beagle Board (see the maroon-colored board in Figure 1), which features the OMAP3530 processor. The processor incorporates a superscalar ARM Cortex-A8 core running at 600MHz as well

NIME'11, 30 May–1 June 2011, Oslo, Norway.

as an a TMS320C64x+ 16-bit fixed-point DSP running at 430MHz. The Beagle Board connects by way of a USB hub incorporating Ethernet to the Arduino Nano, which is inserted into a solderless breadboard (see Figure 1, rear). The following list shows the parts that we used in our initial Satellite CCRMA design with our students during the Autumn quarter in 2010:

- Beagle Board Rev C4[1]
- Arduino Nano[2]
- Solderless breadboard
- 4GB (or larger) SD card
- GWC Technology HE2440 USB 2.0 4-Port Hub with Ethernet Adapter
- Two GT Max adjustable-length USB cables
- Ethernet cable
- 2.5A 5V switching power adaptor (For example DVE DSA-15P-05 US)

Since this deployment, we have begun moving the platform to the similar Beagle Board XM. Depending on the intensity of the output sound and the loudspeaker size required, we suggest supplementing the kit with compact mobile speakers so that sound production can be localized to the instrument.

2.2 Software

We have prepared a special SD card image for the platform that boots Ubuntu Linux into an environment with preinstalled Linux audio applications including the Jack audio server, Pure Data Extended, Faust, JackTrip and ChucK. Because Satellite CCRMA can easily be connected to the Internet, many new packages can be easily installed (e.g. `sudo apt-get install lynx`). Other packages may need to be compiled for the ARM architecture; however, cross-compiling may not be necessary because the SD card installation comes with the gcc tools preinstalled. The SD card image is especially valuable because we have tested it—otherwise, compiling a new SD card image from scratch can easily require multiple days of work. For more information on our SD card image and the community we are building for artists, please see the following link:
`https://ccrma.stanford.edu/~eberdahl/Satellite`

In our class, we currently have students program their sound synthesis engines using Pure Data Extended. This graphical environment allows rapid prototyping by connecting together objects using patch cords. For example, a student can login to his or her Satellite CCRMA kit remotely over Ethernet, load the audio server using the command `qjackctl &`, start the audio server, and finally start Pure Data Extended by typing `pd &`. This causes an X-Window to be forwarded over the Ethernet connection, including something similar to the Pure Data Extended patch shown in Figure 2.

3. MOTIVATION

A generic architectural diagram for musical interaction design platform is shown in Figure 3. Although the specific microcontrollers, computer operating systems, hardware, software, firmware, communications protocols, etc used by different members of the NIME community differ, this diagram

[1]See `http://beagleboard.org/hardware`
[2]See `http://www.arduino.cc/en/Main/ArduinoBoardNano`

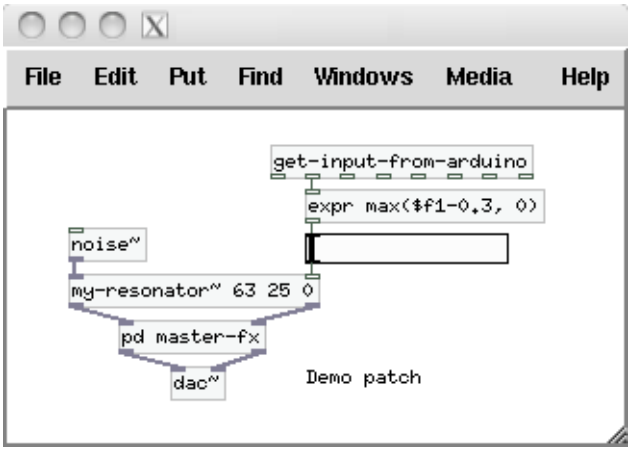


Figure 2: Demonstration patch in Pure Data

reflects the fact that most musical interaction platforms use both a microcontroller (to collect and process sensor data, and also to control actuators) and a microprocessor (to perform more computational challenging tasks such as sound synthesis). What we have observed is that the weak link, as far as longevity and robustness are concerned, is the computer housing the microprocessor. Because computers are expensive, people seldom devote separate computers to their NIME designs; instead they use the same machine that they are writing emails and theses on as the critical engine of their new musical instruments. These instruments thus suffer collateral damage every time we upgrade our operating systems, or install a new version of Java, or switch to new hardware.

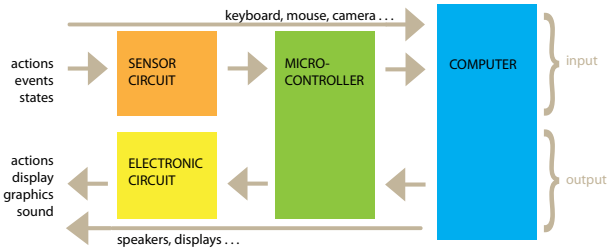


Figure 3: Physical Interaction Design Platform Architecture

At a base level, any system that we would use in our class would need to be affordable (students balk at paying more than $150–what they pay for textbooks in a normal class), adoptable (able to be picked up and put to novel use within a school term), and extensible (able to support a wide variety of ideas that we the instructors didn't think of when we gave the students the systems). We also had the following criteria in mind when we were developing the system:

3.1 Longevity

Most acoustic musical instruments stay operable for many years with minimal maintenance. Hence, as soon as a musician acquires a musical instrument, he or she can often assume that he or she can practice, learn repertoire, compose, and even make adjustments to the instrument for many years to come. Hence, the *longevity* of an acoustic musical instrument promotes the development of virtuosity.

As we mentioned before, we believe that one of the challenges to longevity in prior music interaction platforms was

(a) Quadrofeelia (b) Tub (c) Daft Datum

Figure 4: Some student projects made with Satellite CCRMA

that they relied on the use of a non-dedicated computer to stay within a reasonable price point.

3.2 Independence/Autonomy

One of our objectives in our design was to have the Satellite CCRMA platform be stand-alone. To this end, we set up the operating system so that it could boot reliably without a keyboard, mouse or monitor. We strove to minimize the number of wires that needed to be connected to the system. We particularly focused on eliminating the need for the specialized cables that Beagle Board commonly use. Each additional cable or connector or AC adapter that had to be connected to the system at design or performance time was viewed as a liability, for the misplacement of any piece of the system is an obsolescence risk. Configured with battery power and lithium-ion powered speakers, it is possible to sense, synthesize and generate sound from just the Satellite itself.

We hope that the Satellite CCRMA platform will promote the development of long-lived projects by promoting independence of external systems. This is why the platform incorporates both a microcontroller for sensing and a processor for synthesizing sound. Although it is possible to connect Satellite CCRMA to the internet over either an Ethernet connection or wirelessly, we recommend that designers consider permanently disconnecting final projects from the Internet so that they are as independent as is practical.

3.3 Native Floating Point Computation

These projects require synthesis algorithms to synthesize the sound, but these can be computationally demanding. Furthermore, although it is possible to implement most sound synthesis algorithms using integer computations or fixed-point computations, this process can be challenging and time consuming. Indeed, David Zicarelli once quipped that he had heard that a programmer "could either write fixed-point code or stay married." Such difficulties could distract students from design goals, forcing them to focus on specific engineering problems. Finally, many open-source libraries for synthesizing sound, such as Pure Data Extended, require floating-point computations. For this reason, it was important that the Satellite CCRMA platform was capable of *natively* carrying out *floating-point computations*, as supported by the OMAP3530 processor.

3.4 Compactness

To promote flexibility and integration into other objects, the Satellite CCRMA is *compact*. The Beagle Board itself has a 3"x3.5" footprint, and the Satellite CCRMA board fits into a 5.25"x7"x2" envelope, even with a generate breadboard for electronics onboard. This is small and light enough that it easily could be attached to a violin. Small systems are easier to carry around, which meant that students were more likely to take their projects home to work on. The upcoming beta release of Satellite CCRMA will be even more compact as it is based on the Beagle Board xM with on-board Ethernet and USB support, so the GWC Technology HE2440 hub with Ethernet as shown beneath the Beagle Board in Figure 1 will not longer be required.

Only one of the projects in our course demanded a form factor too small for the Satellite CCRMA. In that project, the students made three tennis-ball sized balls that held Arduinos, a PIC-based sound synthesizer chip, a speaker and a battery. Looking forward, it would be an interesting challenge to see what the minimum size of the Satellite CCRMA platform could be while still using standard connectors.

3.5 Reconfigurability and Extensibility

The open source and open hardware nature of Satellite CCRMA promotes *reconfigurability* and *extensibility*, which we found to be crucial to supporting the wide range of project ideas that our students had. For us instructors, the fact that Linux drivers for Ubuntu were already available for a large number of external USB devices made it possible, for example, to quickly set up the Ethernet/USB hub and the Arduino for our students. For our students, it meant that they could add new hardware (such as bluetooth dongles, web cameras, and multi-channel audio devices) and new software (such as emacs, alpine, cwiid, jwm) with relative ease.

The incorporation of the Arduino Nano and breadboard was also an important aspect to the Satellite CCRMA platform. To some degree, the Arduino was extraneous, because the Beagle Board's expansion port includes multiple serial interfaces and general-purpose I/O lines. However, we felt that it was very useful to be able to leverage our previous Arduino-based curriculum to help our students think about how to design interfaces and controls that were appropriate to the use, context, expression and metaphors they had selected. In the future, we will examine whether the extensibility offered by the Arduino over the native Beagle Board I/O is worth the foot print it takes up.

3.6 Community support

We previously mentioned how leveraging open source hardware and software made the Satellite CCRMA platform easy to reconfigure and extend. The open-source platform also makes it easier to develop a system of community support. Beginning users can leverage examples so that they can get the platform up and running, while advanced users can find out how to tweak the system so that they have more con-

trol over the projects that they develop. For example, the source code for any portion of the software can be obtained, modified, recompiled, and then loaded onto an SD card for use. Similarly, the schematics and layouts for all of the core parts are available, enabling users to make custom boards based on the current state of the platform to meet any special needs.

Inspired by the community-based support and development efforts of platforms such as Processing [13] and Arduino [9], we have created a newsgroup aimed especially at artists to help address questions about the platform: `http://groups.google.com/group/satelliteccrma`. To help bootstrap the learning process, we provided our students with working code examples; these are available to other instructors, students or developers on the Satellite CCRMA site.

4. RELATED WORK

4.1 Prior platforms

In the past, our course has been taught with platforms based on the Basic Stamp [17], the AVRmini [20], and the Arduino [9]. Other past and current microcontroller-based platforms used for creating musical controllers include I-CubeX's Digitizer[11], STEIM's SensorLab and junXionboard, Microchip PIC-based Create USB interface [12], and CNMAT's uosc. All of these systems are meant to be interfaced with a computer, which receives data from the microcontroller (via serial, openSoundControl [21], MIDI formats, for example) to synthesize sounds in software. While these systems all do an admirable job of translating physical phenomena from the real world to signals which can be translated to sound, they require the presence of a laptop or desktop computer to perform real-time sythesis of sound.

4.2 Similar platforms

Far fewer platforms are designed to replace the laptop or desktop computer in the physical interaction architecture. One notable platform is the Audiopint [10] platform. The Audiopint, a 17cm x 17cm "mini-itx" VIA Epia EN1500 motherboard running Ubuntu Linux housed in a Pelican case, functions as a low-cost and flexible audio effects processor and syntehsizer. It uses Pure data [16] to synthesize sound. Another platform is the Gluiph [7], which features Pure data running on a complex programmable logic device (CPLD).

Satellite CCRMA builds on the ideas embodied by the Audiopint and Gluiph. The choice of the Texas Instruments' OMAP-based Beagle Board provides Satellite CCRMA with a small footprint (the Beagle Board itself is 3" wide x3.125" long x0.625" tall, and the platform footprint is 7" x 5.5"x2"), relatively low power consumption, and the potential for broader platform support from the wider OMAP and Beagle Board development community. The cost of Satellite CCRMA, critical to the cost-sensitive artists and musician community, is lower than either platform ($125 at time of submission). Also, the use of a standard Linux operating system makes it easy to extend the system through opensource software and the use of USB-based computer peripherals and drivers available on-line. Finally, the incorporation the bread Board and Arduino helps to orient the use of the system towards the development of novel musical controllers and sound installations.

4.3 Alternative platforms

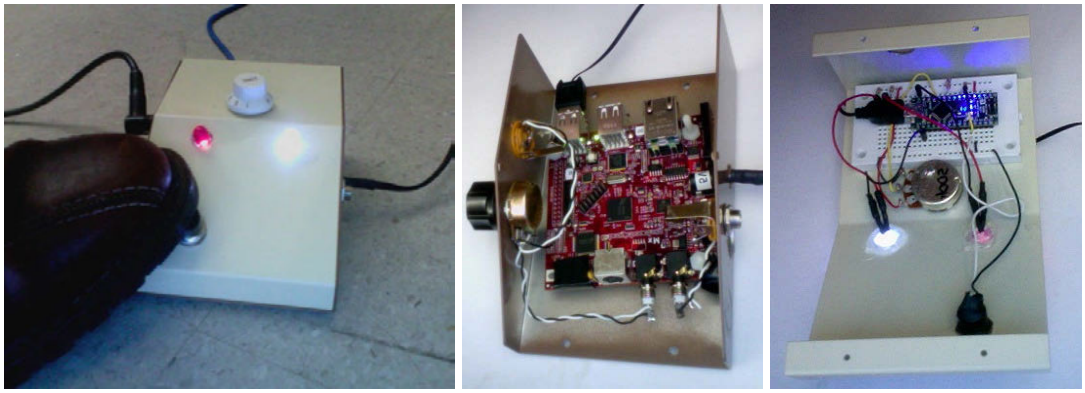
One viable alternative to the Satellite CCRMA system is the use of low-cost netbooks in place of laptops. At the time of this paper's submission, low-end netbooks capable of running Pd cost $300-$400 USD. Although little has been published about the use of netbooks as musical system platforms, we believe it is only a matter of time before people start to dedicate inexpensive netbooks to their instruments. The catch is that, as small as netbooks are, the minimum size and layout of netbooks is dictated by typical laptop usage. Even though the cost of integrating features such as a screen, a lithium-ion battery and wifi and bluetooth capability to the Satellite CCRMA system would likely cost more than a similarly equipped netbook, the cost and difficulty of hacking and reconfiguring a netbook into a format that can be easily incorporated into an instrument would likely neutralize any potential savings.

Another alternative to using the laptop or desktop computer as a music platform is to use mobile phones or iPods. In earlier mobile music platforms, the phones were just used for their autonomous sensing capabilities, and an external PC was used to generate the sound, but more recently, sound synthesis on-the-phone has been pioneered using the Synthesis ToolKit on the Symbian OS [5], and the MoMu on the iPhone OS [2]. The advent of the Android OS phones and tablets is likely to greatly decrease the cost of compact autonomous systems with the computational ability to generate sound, so that these devices will be cheap enough to dedicate to specific instrument or installation designs. However, it is fairly difficult as yet to interface external sensors with these systems, which greatly limits the range of musical control and interaction capabilities of platforms based on these systems. Most strategies involve using Bluetooth wireless, which introduces some latency due to the Bluetooth protocol.

5. PROJECT HIGHLIGHTS

(a) System (b) With top opened

Figure 5: SoundFlinger



(a) Stompbox (b) Bottom FX Pedal (c) Top FX Pedal

Figure 6: Example Stompbox featuring Beagle Board-xM

To illustrate the range of what is can easily be accomplished with Satellite CCRMA as a platform, we present these examples of what our students created for the class projects. Many of the students were new to the Arduino when they started the class, and all of the students were new to the Beagle Board and embedded Linux.

5.1 Independent Musical Instrument

Quadrofeelia, by Jiffer Harriman, Mike Repper, Linden Melvin, and Locky Casey, (see Figure 4(a)) shows that Satellite CCRMA can be employed to construct an *independent* musical instrument that does not require connections to any external computers. A musician plugs the 5V power supply into a wall outlet, and then the 1/4" audio output jack produces an output signal that can be connected to a guitar amplifier or public address system. As the musician manipulates the sensors embedded in the instrument, output sound is produced in response. Quadrofeelia bears resemblance to a slide guitar, but with more abilities for retuning the chord tunings and inversions due to the buttons manipulated by the right hand.

5.2 Independent Dance Interface

Standard sensing devices typically employed for gaming are incorporated into Daft Datum, a musical instrument played by the feet, by Rosie Cima, Ravi Kondapalli, and Ben-Zhen Sung. A dance pad is hidden beneath the performer's feet and sends data to Pure Data over USB using the `hid` object. To discover the vendor and product IDs for the dance pad, the `dmesg` command can be executed in Linux directly after plugging in the dance pad. No further configuration is necessary. To change the sound synthesis mode, the performer presses buttons on the Wiimote as shown in Figure 4(c). The Wiimote communicates with Pure Data over a remarkably small Wifi USB dongle that can be setup as described on the following Wiki: `https://ccrma.stanford.edu/wiki/Making_a_Wii_remote_talk_to_Pure_Data_(PD)`

5.3 Video-based Interactive Installation

The Tüb, by Marc Evans, Björn Erlach, and Mike Wilson, demonstrates the video capabilities of Satellite CCRMA. The KWC-1301 USB Webcam is supended above a tub of water (see Figure 4(b), top center) and transmits images to Pure Data running on the Beagle Board stored underneath the tub. Objects in Pure Data's Graphics Environment for Multimedia (GEM) generate sound samples by scanning along circles in the images. If the water is still, then there is almost no sound, but waves in the water create buzzing sounds, whose timbre evolves with the wave motion. Installation visitors can induce waves in the water using a variety of techniques: poking the water with their hands, pulling the edge of the tub with their hands, poking the water with tubes, blowing water through tubes, etc.

5.4 Multichannel Audio in a Collaborative Installation Piece

The Sound Flinger, by Chris Carlson, Hunter McCurry, and Eli Marschner, demonstrates that Satellite CCRMA is compatible with an external sound interface. The device, as shown in Figure 5, allows users to play back snippets of live recorded sounds through four output audio channels provided by the SIIG IC-710112 USB Soundwave 7.1 Digital audio adapter, which is placed underneath the Beagle Board circled in red in Figure 5(b).

5.5 Audio Effects Stompboxes

Because Satellite CCRMA incorporates audio codecs and native floating-point computation, implementing digital audio effects is relatively straight-forward even for novices. The design history of audio effects incorporates a rich past of audio effects controlled primarily by knobs, switches, and even sometimes pedals. Figures 6 and 6(b) (left) show an example *stompbox*, which incorporates a footswitch and two extra large LEDs. The stompbox demonstrates that Satellite CCRMA can also be employed for teaching audio digital signal processing (DSP) in an embedded systems context. Appropriate languages for DSP include Faust, C++, Pure Data, ChucK, etc.

6. SATELLITE CCRMA WEBSITE

The main website for the project can be found below, which provides links to videos of project presentations, the Satellite CCRMA newsgroup, Wiki, and SD card images: `https://ccrma.stanford.edu/~eberdahl/Satellite/`

7. FUTURE

Since the beginning of this project, we have already had one major hardware change, from the OMAP 3530-based Beagle Board to the OMAP 3730-based Beagle Board-xM. While we believe the OMAP-based Beagle Board family to be a promising platform for the design of new interfaces for musical expression, the core aim of the Satellite CCRMA project is to develop standalone computational capabilities for musical interfaces in an open-source setting. As such, we are tracking related platforms such as the Hawk Board, Panda Board, Crane Board, or GumStix, in case any of these should become more viable for our project goals. In

any case, it is our intent to maintain a similar set up of core platform software across hardware changes, and also to make migration from one platform to the next graceful for users.

We have found that the existing platform is stable and easy enough for NIME students to use, reconfigure and extend within the confines of a single school term. In the future, we hope to see if the Satellite platform helps promising projects grow beyond their short-term novelty "stunt" phase to become more mature musical instruments. In particular, we hope to see whether inventors develop more expertise with their designed installations and instruments, perhaps by making more refinements, developing greater expertise in playing their instrument, or by making scores for the systems' unique capabilities. We also intend to perform more detailed analysis to see if the Satellite CCRMA system helps students to more fully understand full-scale system design.

8. CONCLUSIONS

In conclusion, we found that the introduction of the Satellite CCRMA platform has provided our students with a more full-featured, robust and flexible system for lab exercises and musical interaction design projects. This platform extends the portability, scalability and support community of our previous microcontroller-based system to the microprocessor as well. We have found the platform is useful for instrument-based and installation-based systems, and we are actively promoting the platform to see if it is useful in a wide variety of other applications as well.

9. ACKNOWLEDGMENTS

In addition to Chris Chafe, Fernando Lopez-Lezcano, Bill Verplank, Max Mathews, Perry Cook, Julius Smith III, Michael Gurevich, Carr Wilkerson, and the open-source community, we would like to thank our students who helped us test the initial release of Satellite CCRMA: Chris Carlson, Locky Casey, Roseann Cima, Björn Erlach, Marc Evans, Francesco Georg, Jiffer Harriman, Ravi Kondapalli, Eli Marschner, Hunter McCurry, Linden Melvin, Michael Repper, Mike Rotondo, Spencer Salazar, Ben-Zhen Sung, and Mike Wilson. This research was made possible by a generous equipment donation from Texas Instruments.

10. REFERENCES

[1] A. Bowen. Soundstone: a 3-d wireless music controller. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 268–269. National University of Singapore, 2005.

[2] N. J. Bryan, J. Herrera, J. Oh, and G. Wang. MoMu: a mobile music toolkit. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME), Sydney, Australia*, 2010.

[3] J. Carlile and B. Hartmann. OROBORO: a collaborative controller with interpersonal haptic feedback. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 250–251. National University of Singapore, 2005.

[4] L. Dahl, N. Whetsell, and J. V. Stoecker. The WaveSaw: a flexible instrument for direct timbral manipulation. In *Proceedings of the 7th international conference on New interfaces for musical expression*, pages 270–272. ACM, 2007.

[5] G. Essl and M. Rohs. Mobile stk for symbian os. In *Proc. International Computer Music Conference*, pages 278–281. Citeseer, 2006.

[6] M. Gao and C. Hanson. LUMI: live performance paradigms utilizing software integrated touch screen and pressure sensitive button matrix. In *Proceedings of the 2009 Conference on New Interfaces for Musical Expression*, 2007.

[7] S. Kartadinata. The gluiph: a nucleus for integrated instruments. In *Proceedings of the 2003 conference on New Interfaces for Musical Expression*, page 180. Citeseer, 2003.

[8] R. Lugo and D. Jack. Beat boxing: expressive control for electronic music performance and musical applications. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 246–247. National University of Singapore, 2005.

[9] D. Mellis, M. Banzi, D. Cuartielles, and T. Igoe. Arduino: An open electronic prototyping platform. In *Proc. CHI, 2007*, 2007.

[10] D. Merrill, B. Vigoda, and D. Bouchard. Audiopint: A robust Open-Source hardware platform for musical invention. *Pd Convention*, 2007.

[11] A. Mulder. The I-Cube system: moving towards sensor technology for artists. In *Proc. of the Sixth Symposium on Electronic Arts (ISEA 95)*, 1995.

[12] D. Overholt. Musical interaction design with the create usb interface. In *Proc. ICMC2006, International Computer Music Conference, New Orleans*, 2006.

[13] C. Reas and B. Fry. Processing: a learning environment for creating interactive web graphics. In *ACM SIGGRAPH 2003 Web Graphics*, page 1. ACM, 2003.

[14] D. Schlessinger and J. O. Smith. The kalichord: A physically modeled Electro-Acoustic plucked string instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 98–101, 2009.

[15] H. Shiraiwa, R. Segnini, and V. Woo. Sound kitchen: designing a chemically controlled musical performance. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 83–86. National University of Singapore, 2003.

[16] H. C. Steiner. Building your own instrument with pd. In *Proceedings of the 1st International Pd Conference, Graz, Austria*. Citeseer, 2005.

[17] B. Verplank, C. Sapp, and M. Mathews. A course on controllers. In *Proceedings of the 2001 conference on New interfaces for musical expression*, pages 1–4. National University of Singapore, 2001.

[18] C. Wilkerson, C. Ng, and S. Serafin. The mutha rubboard controller. In *Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–4. National University of Singapore, 2002.

[19] S. Wilson, M. Gurevich, B. Verplank, and P. Stang. Microcontrollers in music HCI instruction: reflections on our switch to the atmel AVR platform. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 24–29. Citeseer, 2003.

[20] S. Wilson, M. Gurevich, B. Verplank, and P. Stang. Microcontrollers in music HCI instruction: reflections on our switch to the atmel AVR platform. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 24–29. Citeseer, 2003.

[21] M. Wright and A. Freed. Open sound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the 1997 International Computer Music Conference*, pages 101–104, 1997.

Two Turntables and a Mobile Phone

Nicholas J. Bryan and Ge Wang
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
660 Lomita Dr.
Stanford, California, USA
{njb, ge}@ccrma.stanford.edu

ABSTRACT

A novel method of digital scratching is presented as an alternative to currently available digital hardware interfaces and time-coded vinyl (TCV). Similar to TCV, the proposed method leverages existing analog turntables as a physical interface to manipulate the playback of digital audio. To do so, however, an accelerometer/gyroscope–equipped smart phone is firmly attached to a modified record, placed on a turntable, and used to sense a performers movement, resulting in a wireless sensing-based scratching method. The accelerometer and gyroscope data is wirelessly transmitted to a computer to manipulate the digital audio playback in real-time. The method provides the benefit of digital audio and storage, requires minimal additional hardware, accommodates familiar proprioceptive feedback, and allows a single interface to control both digital and analog audio. In addition, the proposed method provides numerous additional benefits including real-time graphical display, multi-touch interaction, and untethered performance (e.g "air-scratching"). Such a method turns a vinyl record into an interactive surface and enhances traditional scratching performance by affording new and creative musical interactions. Informal testing show this approach to be viable, responsive, and robust.

Keywords

Digital scratching, mobile music, digital DJ, smartphone, turntable, turntablism, record player, accelerometer, gyroscope, vinyl emulation software

1. INTRODUCTION

The performance practice of DJing has experienced astonishing growth over the past three decades. Scratching, beat-matching, beat juggling, mixing, and similar techniques can be heard on the radio, in night clubs, and experimental music contexts around the world. All such performance styles can be traced back to the unique physical interaction and expressive nature of a simple mechanical device–the analog turntable. The simple physical control and inherent proprioceptive feedback affords the possibility of incredible virtuosity and skill without hindering a beginner's zeal.

With the advent of digital audio, however, great attention has been focused on digital implementations of the turntable so as to leverage the many benefits of digital storage and playback. Such implementations typically fall within two categories: methods leveraging existing analog turntables with certain modification and methods requiring alternative hardware mimicking the turntable's control. Examples of the prior include time-coded vinyl (TCV), while examples of the latter include CDJs or similar interfaces [2, 1, 4, 3]. TCV uses a vinyl record encoded with time-code to detect needle position, where as alternative hardware uses various alternate sensing mechanisms. Both approaches have distinct advantages and disadvantages, largely dependent on personal preference and performance style.

In many traditional DJ settings, time-coded vinyl methods have proven overwhelmingly popular. TCV methods allow a single interface to control both analog and digital audio and maintain the familiar and nimble scratching control of a traditional analog turntable. Disadvantages include wear and tear on the vinyl record, limited duration of the TCV records, possible jumps in needle position during a performance, and physical interference of the turntable tone arm.

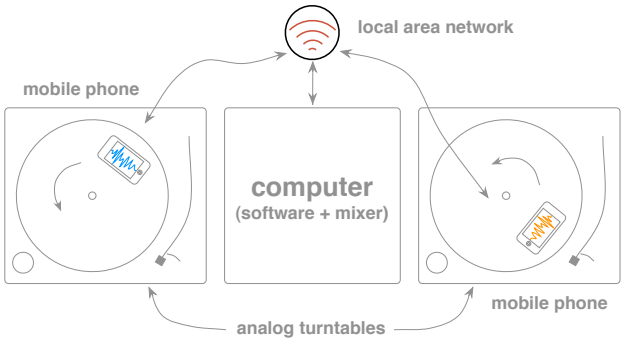


Figure 1: Proposed DJ Setup. The system uses turntables and sensor-equipped mobile phones, networked with a host computer.

In this work, a novel method of digital scratching is presented as a viable alternative to currently available digital hardware interfaces and TCV. The proposed method leverages existing analog turntables as a physical interface to manipulate the playback of digital audio, but does not require a time-coded record or any physical connection to a digital audio playback device. An accelerometer/gyroscope–equipped smart phone atop a modified record is used to wirelessly transmit gesture data to a computer and manipulate the digital audio playback accordingly as shown in Fig. 1. Using modern smart phones as a prototyping platform gives similar benefits as TCV, but provides numerous additional benefits including added visual display, multi-touch interaction, and untethered performance (e.g "air-scratching"),

NIME'11, 30 May–1 June 2011, Oslo, Norway.

turning a vinyl record into a familiar, but new tangible interface [19]. Informal testing shows promising results, with minimal latency and comparable feel when matched against alternative approaches[1].

2. RELATED WORK

Alternatives to commercially available digital DJ methods include various form factors and implementations. Bill Verplank references former students Keatly Halderman, Daniel Lee, Steve Perella and Simon Reiff replacing an existing phonograph needle with a riding wheel equipped with an optical shaft encoder to digitize the gesture control [30]. Hans and others developed DJammer, an accelerometer equipped MP3 player [27, 12, 13, 14]. In addition to untethered control of DJ gestures, DJammer presented the use of virtual jam sessions to exchange and share music audio streams. Sile O'Modhrain discusses the importance of haptic feedback for musical instrument design in [24] and emphasizing such importance, Beamish created the D'Groove haptic turntable device in [5].

The use of wearable sensors for real-time music signal processing is presented in [21], while an overview of designing alternative tools for turntable music in the digital era is presented in [22]. Villar et al. introduced the ColorDex DJ System [31], using color as a mixing metaphor. Hansen et al. have done extensive work on the acoustics and performance of scratching [17, 15] as well as high level gesture control [16, 18].

Multi-touch interfaces have also been used within many musical and DJ applications including the popular Reactable [20] and numerous commercially available DJ applications. A gesture based mobile music game involving touch-screen scratching was presented [11]. Most recently, Savage et al. introduced a multi-modal mobile music mixer using mobile phones with accelerometer for gesture control of Bluetooth streaming audio [26].

Surveying such past academic work emphasizes the various important differences in approach when designing new turntable interfaces. Firstly, there is a distinction between interfaces which are meant to enhance the turntable performance experience while maintaining the traditional physical interaction of manipulating motor movement versus interfaces which are meant to alter the interaction with equivalent audio effect. For better or worse, the goals between the two approaches are notably different. Secondly, there is a difference between the DJ performance practices of mixing and scratching. A large number of alternative interfaces focus on mixing because of the latency and sensitivity requirements. Scratching, described as the art of manipulating a vinyl record against a turntable needle, as well as the related scribbling (rapid scratching) [17] requires accurate, low-latency, highly-sensitivity sensing.

3. APPROACH

Within the various approaches found in recent and past research, we present work towards the *enhancement* of the turntable performance using existing analog turntable hardware. In addition, we focus on digital scratching interaction. As a general approach, we begin by leveraging the portability and computing power of modern mobile phones (or smartphones), which have been shown to provide a highly expressive compact form factor [32, 23]. Alternative sensors such as a wirelessly enabled light sensors or similar have the ability to offer a more compact form factor for rotation sensing, but do not provide numerous other advantages provided by modern smartphones.

[1] `http://ccrma.stanford.edu/~njb/research/turntable/`

3.1 Accelerometer and Gyroscope Sensing

Accelerometer and gyroscope–equipped smartphones, in particular, can be used to sense and wirelessly transmit gestural control data. With proper processing, a three-axis accelerometer and gyroscope can detect three-axis rotation rate (pitch, roll, and yaw velocities) ideal for sensing motion on a turntable. As a result, by firmly attaching a properly equipped mobile phone atop a vinyl record, an existing analog turntable can easily be modified into a digital scratching interface requiring no specialized sound card. Such a method maintains a near equivalent sense of tactility and results in a wireless sensing-based scratching method as seen in Fig. 2. Further, the wireless sensing method does not

Figure 2: Wireless Sensor Record In Action. A prototype record (combination of mobile phone, sticky rubber, and plexiglass disc) resting on a standard, commercially available turntable.

have any length limitation as found with TCV and advantageously avoids physical interference of the turntable tone arm.

For many situations, the capabilities of accelerometers and gyroscopes are not ideal. The processing required is non-trivial and demands careful attention. Small errors in acceleration measurements propagate to larger errors in velocity and position estimates, commonly referred to as drift or bias. Gyroscopes, however, provide a complementary measure of orientation and can be used to improve accelerometer measurements via complimentary filters, statistical filters (i.e. Kalman filters), or other methods collectively referred to as sensor fusion algorithms. In addition, physical constraints can be added to further improve estimates such as limiting the axes of measurement. Serendipitously, the motion of scratching gestures are limited around a single axis and even more so, the motion is circular, directly relating centripetal force (provided by the performer) to rotational velocity. As a result, the use of accelerometer and gyroscope–equipped smartphones for precisely sensing scratching gestures is surprisingly suitable.

3.2 Proposed Method

By processing the continuous data stream of the accelerometer and gyroscope, the system can achieve a precise and robust measurement of instantaneous rotational velocity. This allows us to robustly track both steady and variable rotational velocity. Remarkably, this works well even for more extreme changes in rotational velocities such as those produced by the physical gesture of scratching. By transmitting this data over a low-latency wireless network, the physical gestures applied to the mobile phone can be mapped to

the playback position of an audio file or, more generally, any other real-time audio parameter. This creates a viable alternative to prior digital scratching methods.

4. INTERACTIONS

The proposed method enables a number of additional benefits and novel interactions, taking advantage of multi-touch displays as well as the physicality and mobility of the modern mobile phone.

4.1 Visual Feedback

By placing a mobile phone on top of the moving vinyl record and adding real-time "on-record" visual feedback with multi-touch interaction, a simple vinyl record is transformed into an interactive surface. This can aid a performer in numerous contexts including cueing, scratching, and beat juggling. The processing of cueing involves preparing one record to mix in with another and involves matching tempo, musical phrasing, or similar musical properties. Direct visual feedback on the record can help in this process, even suggesting how to modify the speed controls of the analog turntable.

When performing scratching and beat juggling, DJs will typically place a visual marker (e.g., tape, paint, Post-it, etc.) to remember the playback point within a certain song as described in [5]. Having on-record visual feedback can directly aid in this process. Fig. 3 shows an example implementation displaying the actual audio signal itself on the moving record, visually displaying the current position within a song. As the record moves, the visual display is updated according to the exact position of the audio file playing on the host computer with the window size or length of the displayed audio controlled using multi-touch pinch-to-zoom controls. As seen, the start of a percussive sound

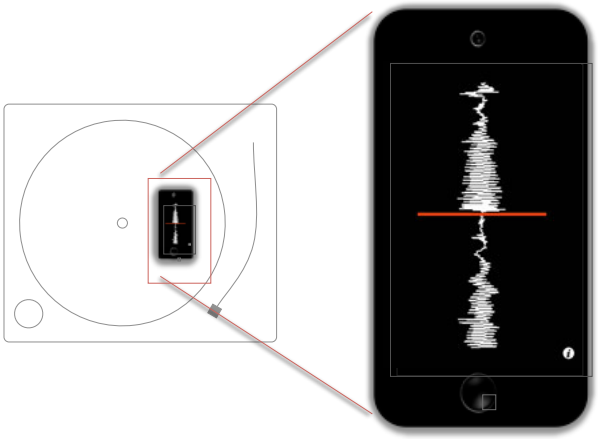


Figure 3: On-Record Visual Feedback. An example of on-record visual feedback displaying the time-domain audio waveform via custom software.

is displayed indicating a possible physical location within the record for scratching or beat juggling. More detailed visualizations could display virtual paint markings, tape, or Post-its for a familiar style indication or entirely different information and graphical user-interfaces. In addition, a performer could use such visual feedback to select the "needle" position within a song or even switch between multiple songs. As multi-touch technology advances, one could image a multi-touch display covering the entire record surface.

4.2 Gesture Modification

By using digitized gesture control, alternative gesture-to-sound mappings are possible. As standard in digital DJing, this can be found in the form of independent sensitivity, pitch, and tempo control. More specifically, scratching gesture can be *amplified* or *dampened* by scaling the transmitted rotational rate accordingly.

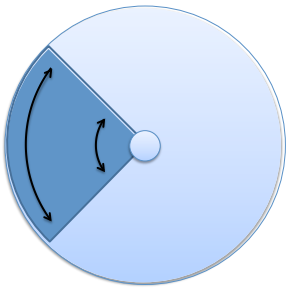


Figure 4: Changing Tone Arm Position Affects Gesture Sensitivity. The changing tone arm position affects scratching gestures as a function of line area.

As an alternative interpretation, sensitivity control can be seen as controlling the tone arm position within a record. Traditional analog turntables operate with constant velocity rotation, forcing audio material on the inner grooves of a record to correspond to proportionally longer length signals for an equivalent angle as seen in Fig. 4. This causes an exactly repeated scratching gesture to sound differently depending on the tone arm position. This effect can either be replicated or removed depending on personal preference.

In addition, gesture sensing can be set relative to still or constant motion. This allows the system to be used with or without the turntable motor in action. The measured rotation speed can simply be biased so still motion corresponds to a playback rate of 1.0 instead of 0.0, allowing the performer to choose his or her preference. This is not possible with traditional TCV, which requires active rotation.

Finally, as presented in [25, 8, 9] and others, such gestural control can also be used for *active listening*, allowing the general public of listeners and inexperienced users interact with the music generation and manipulation, not just trained musicians.

4.3 Untethered Scratching

While serving the purpose of enhancing traditional scratching gestures tethered to an existing analog turntable, the presented approach affords alternative interactions that can lead to new forms of expression. In particular, untethered or "air" scratching can be performed by simply lifting the mobile phone-equipped record off the turntable as no physical sensor connections are required. As discussed in [27] and [26], untethered interaction frees the performer to move about and even interact directly with the audience. Such ability poses numerous interesting questions regarding improvisation techniques and other musical devices. Involving audience participation during a live scratch performance, for example, is an appealing direction of study.

5. IMPLEMENTATION

For implementation, custom software for both the sensing smartphone and host computer was needed with minimal custom hardware. More detailed hardware and software implementation issues are discussed in §5.1 and §5.2 respectively.

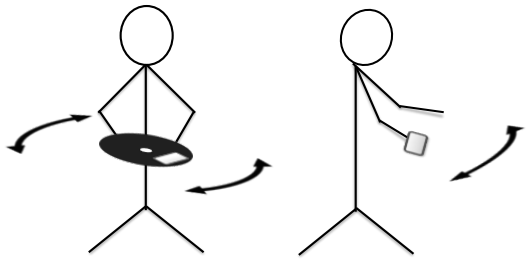


Figure 5: Untethered Scratching Interaction. "Air-Scratching" is possible with or without a physically attached record.

5.1 Hardware

For hardware, a single mobile phone, piece of sticky rubber, and plexiglass disc were used for each wirelessly enabled record. Fourth generation iPod Touch or iPhone 4 devices were found to work well. Both devices include a three-axis accelerometer and three-axis gyroscope with a maximum sample rate of 100 Hz as well as a multi-touch display and wireless networking capabilities. The iPod Touch is physically thinner than the iPhone 4 and found to be slightly preferable. In order to firmly attach the device to a vinyl record, sticky rubber is placed in between the record and mobile device as shown in Fig. 6. Commercially available rubber mats (used to hold mobile phones against a car dashboard) were used and found to be sufficiently sticky.

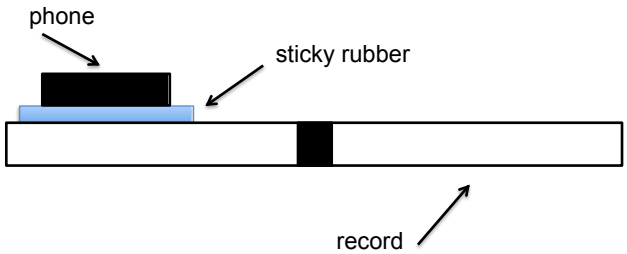


Figure 6: Wireless Sensor Enabled Record. A mobile phone attached to a modified record via sticky rubber.

Various plexiglass discs were used in place of a vinyl record. By varying the weight and size of record, a performer can customize the drag or friction between the record and slipmat to suit their needs. A collage of prototype images is found in Fig. 7, showing a single record with phone and rubber, collection of various discs, and the complete DJ setup.

5.2 Software

Software implementation came in two forms: software on the mobile phone and host computer software. The mobile phone software was written within Apple's iOS SDK along with portions of the Mobile Music Toolkit [7], oscpack [6], and the Synthesis Toolkit [10]. The iOS CoreMotion framework gives an excellent mechanism to stream processed accelerometer and gyroscope data by an unspecified sensor fusion algorithm (most likely complementary or Kalman filtering), directly providing three-axis rotational rates of pitch, roll, and yaw. The yaw rate is then wirelessly transmitted using Open Sound Control on top of UDP sockets. Track information and other non-real-time information can be sent reliably over TCP sockets.

Figure 7: Prototype Hardware. (Upper Left) Phone, plexiglass disc, and sticky rubber. (Upper Right) Various sized and weighted discs to accommodate a performer's sense of tactility. (Lower) Prototype setup.

To adequately implement the visual feedback of the currently playing audio stream as discussed in §4.1, additional information including the position within the currently playing audio file must be sent from the host computer back to the mobile phone to update the display. If the visual display is not updated by the host computer, the two devices will drift from one another causing the audio and visuals to become out-of-sync. Such effect is confusing to the performer and is greatly undesirable.

Host software employs the Jules' Utility Class Extensions (JUCE) [29] providing a cross-platform framework with numerous tools ready for audio application development. A traditional DJ software model is taken, allowing two simultaneous audio streams. The host program receives the transmitted rotational rate and manipulates the audio stream by resampling. Linear interpolation was initially used for prototyping, with improvements found when using higher-order polynomial interpolation [28]. An image of the developed software is shown in Fig. 8.

Figure 8: Custom Prototype DJ Software. The developed DJ software required to receive and processing the mobile phone sensor data and manipulate audio accordingly.

6. DISCUSSION AND EVALUATION

As discussed earlier, there are various advantages and disadvantages when comparing different digital scratching methods. In general, however, it is difficult to objectively evaluate new methods and compare against past work. Just as a violinist gets a custom to the feel and sound of their instrument, a DJ will learn the subtleties of a given scratching method and can be averse to change [5]. Informal performance testing, nevertheless, showed promising results with minimal perceived latency between input gesture data and output audio playback.

General measures of evaluation included precision and responsiveness as well as stability. Rapid physical gestures were seen to be very responsive and have precise corresponding audio effect. Repeated physical gestures were also found to have a consistent sounding effect over long performance times. Further testing with professional level DJs, however, is needed for a more complete evaluation. For a video demonstration of the system in action please see `http://ccrma.stanford.edu/~njb/research/turntable/`.

The one-way network latency time between a given phone and host computer was measured to be on average 3-5 ms and compares favorably with professional audio recording equipment. When comparing the maximum humanly-possible scratch rate (10-20 turns per second [17]), the 100 Hz sample rate of the accelerometer and gyroscope appears suitable. The perceived effect of accelerometer and gyroscope latency, however, is difficult to measure and dependent on the sensor filtering method used, requiring further study and user evaluation.

7. CONCLUSIONS

A straightforward and surprisingly effective method of digital scratching is presented. The proposed method leverages existing analog turntables as a physical interface and takes advantage of the capabilities of modern sensor–equipped smartphones, resulting in a genuinely physical, wireless sensing-based scratching method. Benefits include digital audio and storage, minimal additional hardware, familiar proprioceptive feedback, and a single interface to control both digital and analog audio. Further benefits include visual display, gesture modification, and the possibility of interactions untethered from the turntable. Testing and evaluation show this approach to be viable and promising.

8. ACKNOWLEDGMENTS

This work was enabled by National Science Foundation Creative IT grant No. IIS-0855758 as well as the funding from the School of Humanities and Sciences, Stanford University. Additional thanks to Professor Jonathan S. Abel for valuable conversation regarding the tone arm control. Finally, a thank you to the anonymous reviewers for valuable feedback regarding the application of active listening and track switching, among other observations.

9. REFERENCES

[1] Ms. Pinky, January 2011. `http://www.mspinky.com/`.

[2] Native Instruments, January 2011. `http://www.native-instruments.com/`.

[3] Rane, January 2011. `http://www.rane.com`.

[4] Stanton, January 2011. `http://www.stantondj.com/`.

[5] T. Beamish. D'Groove - a novel digital haptic turntable for music control. Master's thesis, UBC, 2004.

[6] R. Bencina. oscpack, Nov. 2006. `http://www.audiomulch.com/~rossb/code/oscpack/`.

[7] N. J. Bryan, J. Herrera, J. Oh, and G. Wang. MoMu: A mobile music toolkit. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Sydney, Australia, 2010.

[8] A. Camurri, C. Canepa, and G. Volpe. Active listening to a virtual orchestra through an expressive gestural interface: the orchestra explorer. In *Proceedings of the 7th international conference on New interfaces for musical expression*, NIME '07, pages 56–61, New York, NY, USA, 2007. ACM.

[9] A. Camurri, G. Volpe, H. Vinet, R. Bresin, M. Fabiani, G. Dubus, E. Maestre, J. Llop, J. Kleimola, S. Oksanen, V. Välimäki, and J. Seppanen. User-centric context-aware mobile applications for embodied music listening. In O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, M. Gerla, H. Kobayashi, S. Palazzo, S. Sahni, X. S. Shen, M. Stan, J. Xiaohua, A. Zomaya, G. Coulson, P. Daras, and O. M. Ibarra, editors, *User Centric Media*, volume 40 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 21–30. Springer Berlin Heidelberg, 2010.

[10] P. R. Cook and G. P. Scavone. The Synthesis ToolKit (STK). In *Proceedings of the International Computer Music Conference*, Beijing, China, 1999.

[11] N. Gillian, S. O'Modhrain, and G. Essl. Scratch-off: A gesture based mobile music game with tactile feedback. In *Proceedings of the 2009 conference on New Interfaces for Musical Expression*, NIME '09, pages 308–311, 2009.

[12] M. Hans, A. Slayden, M. Smith, B. Banerjee, and A. Gupta. Djammer: a new digital, mobile, virtual, personal musical instrument. *Multimedia and Expo, IEEE International Conference on*, 0:4 pp., 2005.

[13] M. C. Hans and M. T. Smith. Interacting with audio streams for entertainment and communication. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 539–545, New York, NY, USA, 2003. ACM.

[14] M. C. Hans and M. T. Smith. A wearable networked MP3 player and "turntable" for collaborative scratching. *Wearable Computers, IEEE International Symposium*, 0:138, 2003.

[15] K. F. Hansen. *The acoustics and performance of DJ scratching*. PhD thesis, KTH Royal Institute of Technology, 2010.

[16] K. F. Hansen, M. Alonso, and S. Dimitrov. Combining dj scratching, tangible interfaces and a physics-based model of friction sounds. In *Proceedings of the International Computer Music Conference*, pages 45–48, 2007.

[17] K. F. Hansen and R. Bresin. Analysis of a genuine scratch performance. In *In Proceedings of the Gesture Workshop*, pages 519–528, 2003.

[18] K. F. Hansen and R. Bresin. The skipproof virtual turntable for high-level control of scratching. *Comput. Music J.*, 34:39–50, June 2010.

[19] H. Ishii and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '97, pages 234–241, New York, NY, USA, 1997. ACM.

[20] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reacTable. In *In Proceedings of the International Computer Music Conference (ICMC 2005*, pages 579–582, 2005.

[21] A. Kapur, E. Yang, A. Tindale, and P. Driessen. Wearable sensors for real-time musical signal processing. In *Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on*, pages 424 – 427, August 2005.

[22] T. M. Lippit. Turntable music in the digital era: designing alternative tools for new turntable expression. In *Proceedings of the 2006 conference on New interfaces for musical expression*, NIME '06, pages 71–74, Paris, France, France, 2006. IRCAM, Centre Pompidou.

[23] J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang. Evolving the mobile phone orchestra. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2010.

[24] M. S. O'Modhrain. *Playing by feel: incorporating haptic feedback into computer-based musical instruments*. PhD thesis, Stanford University, Stanford, CA, USA, 2001. AAI3000074.

[25] R. Rowe. *Interactive music systems: machine listening and composing*. MIT Press, Cambridge, MA, USA, 1992.

[26] N. S. Savage, S. R. Ali, and N. E. Chavez. Mmmmm: A multi-modal mobile music mixer. In *Proceedings of the 2010 conference on New Interfaces for Musical Expression*, NIME '10, pages 395–398, 2010.

[27] A. Slayden, M. Spasojevic, M. Hans, and M. Smith. The djammer: "air-scratching" and freeing the dj to join the party. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI '05, pages 1789–1792, New York, NY, USA, 2005. ACM.

[28] J. O. Smith. *Physical Audio Signal Processing*. W3K Publishing, 2010. `http://books.w3k.org`.

[29] J. Storer. Jules' Utility Class Extensions (JUCE), Raw Material Software, January 2011. `http://www.rawmaterialsoftware.com/juce.php`.

[30] B. Verplank. A course on controllers. *NIME Workshop at AIGCHI*, 2001. `www.billverplank.com/ControllersCourse2.pdf`.

[31] N. Villar, H. Gellersen, M. Jervis, and A. Lang. The colordex dj system: A new interface for live music mixing. In *Proceedings of the 2007 conference on New Interfaces for Musical Expression*, NIME '07, pages 264–269, 2007.

[32] G. Wang, G. Essl, and H. Pentinnen. Do mobile phones dream of electric orchestras? *Proceedings of the International Computer Music Conference*, 2008.

MadPad: A Crowdsourcing System for Audiovisual Sampling

Nick Kruge
Stanford University
CCRMA
660 Lomita Ct
Stanford, California USA
nkruge@ccrma.stanford.edu

Ge Wang
Stanford University
CCRMA
660 Lomita Ct
Stanford, California USA
ge@ccrma.stanford.edu

ABSTRACT

MadPad is a networked audiovisual sample station for mobile devices. Twelve short video clips are loaded onto the screen in a grid and playback is triggered by tapping anywhere on the clip. This is similar to tapping the pads of an audio sample station, but extends that interaction to add visual sampling. Clips can be shot on-the-fly with a camera-enabled mobile device and loaded into the player instantly, giving the performer an ability to quickly transform his or her surroundings into a sample-based, audiovisual instrument. Samples can also be sourced from an online community in which users can post or download content. The recent ubiquity of multitouch mobile devices and advances in pervasive computing have made this system possible, providing for a vast amount of content only limited by the imagination of the performer and the community. This paper presents the core features of MadPad and the design explorations that inspired them.

Keywords

mobile music, networked music, social music, audiovisual, sampling, user-generated content, crowdsourcing, sample station, iPad, iPhone

1. INTRODUCTION

MadPad is a social music and video creation system currently implemented for the Apple iPad and iPhone. The performer loads twelve independent video clips onto twelve virtual pads (Figure 1) which are laid out in a grid pattern similar to the sixteen pads of an Akai MPC-2000[1] (Figure 2). Upon tapping any pad, the associated audio and video play under the performer's fingertips. Up to eleven can be played simultaneously, and two finger drag gestures can be used to control the playback rate of any individual clip, allowing for expressive control of the content beyond basic re-triggering.

MadPad is a platform that employs the creativity of its users to make it come alive. The application is designed to be a transparent conveyance of user content where the content *is* the instrument. Clips can be created on a camera-enabled mobile device in either rapid succession or one-by-one. In the rapid mode, the performer can simply make twelve separate sounds and they will be automatically distributed to the twelve slots, while the one-by-one mode allows for a more tailored approach, giving the performer as many takes as necessary to capture each desired sample individually.

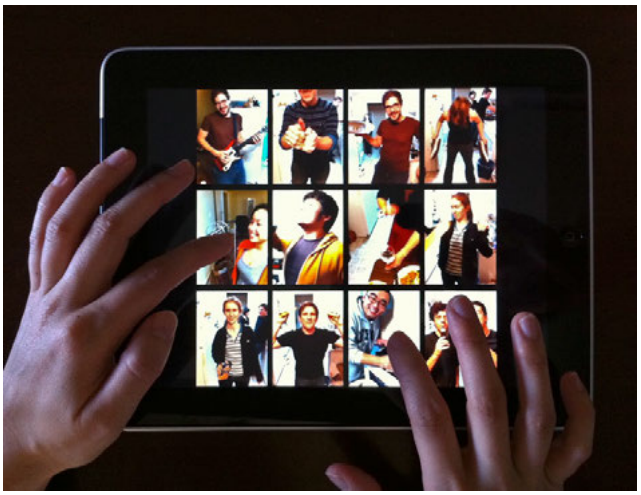


Figure 1: MadPad in action on the Apple iPad. The performer taps on the video clips to play the associated audio and video.

Figure 2: The Akai MPC.

With the ease of creating a sample set in under a minute and the addition of video to the traditional MPC-like sampling paradigm, this system intends to give its user the feeling that there is a potential for music all around, and an instrument can be created out of anything in sight. Furthermore, these sample sets serve as a bridge between still

NIME'11, 30 May–1 June 2011, Oslo, Norway.

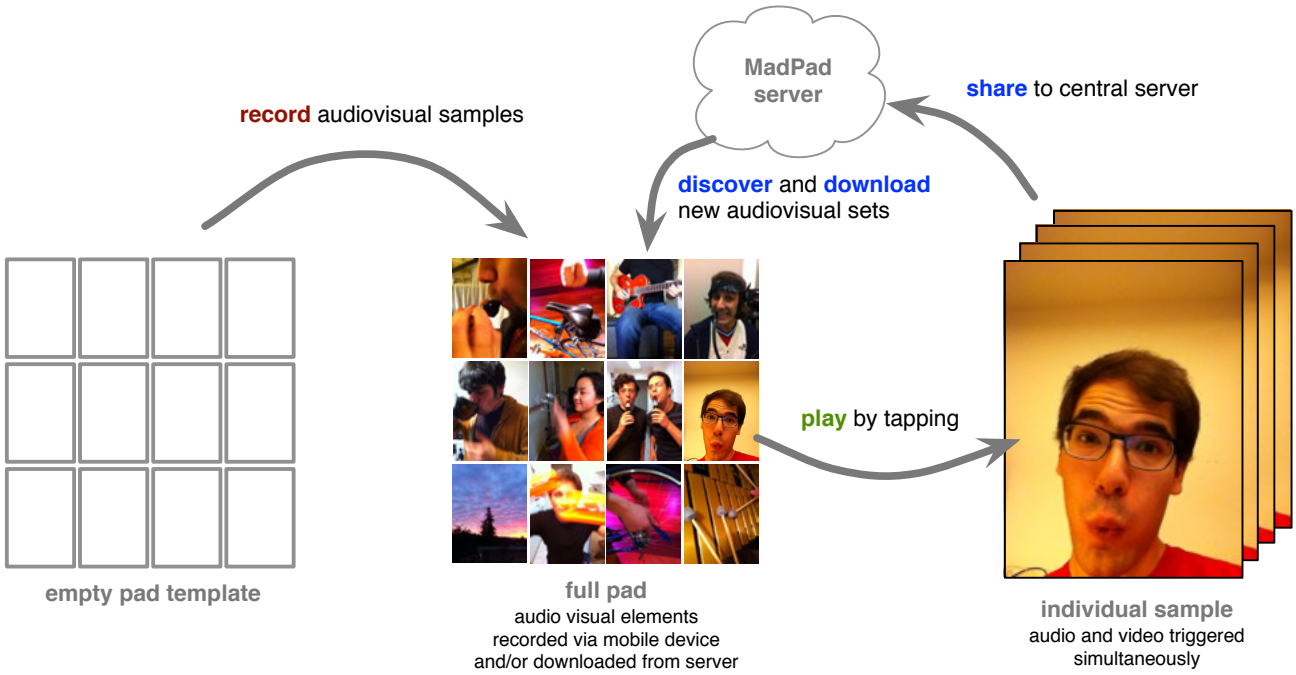


Figure 3: Overview of the MadPad system.

photography and video, allowing moments to be captured, shared, and relived in a novel and interactive way. Once created, there is an online forum that allows users to share their creations with others, enabling a rich community of user-generated content to develop. All of MadPad's features can be experienced by a performer with a single camera-enabled iPad or iPhone connected to the internet, making the entire experience very portable.

2. RELATED WORK

Figure 4: Frame from a VideoSong: Featuring Pomplamoose and Ben Folds

The proliferation of mobile music technology[7] as well as the increasing number of performance outlets for mobile musicians[14] have set the stage for MadPad. A number of existing works, both academic and artistic, have influenced the design and implementation.

MadPad shares aesthetic similarities with a style of music video called the "VideoSong" (Figure 4), which employs repeatedly triggered video samples in multiple panes for effect. Just as in MadPad, these video samples are literal depictions of the associated audio, and display the actual recording of the sound the listener is hearing. In 2006, Norwegian artist Lasse Gjertsen released *Amateur*[9], an audiovisual piece that reuses a handful of tightly cut single drum and piano hit videos in what he refers to as his "hyperactive editing style" to create a full song. In 2009, *Pomplamoose*, an indie rock duo from the San Francisco Bay Area, sold roughly 100,000 songs thanks to several viral online videos [16] and coined the term VideoSong. Although Pomplamoose tends to use longer cuts and more varied layouts than Gjertsen or MadPad, the visual aesthetic is similar. Jack Conte of Pomplamoose defines it with two rules: 1.) What you see is what you hear. 2.) If you hear it, at some point you see it[4].

The basic user interaction of MadPad draws from the Akai Music Production Center (Figure 2), commonly referred to by the acronym *MPC*. The main interaction of the MPC uses 16 finger pads to trigger single audio samples when tapped. With this tool, a larger audio clip can be chopped up quickly and distributed to the pads as different subsets of the original sound[1]. Also, individual and potentially unrelated sound clips can be loaded and triggered singularly. These possibilities combined allow for a large number of sonic sample sets to be formed even with just a few seconds of initial material, giving performers a quick way to move from sound clips to expressive, playable instruments[12]. MadPad uses the large, multitouch display of the iPad to offer this same interaction with videos in place of the pads. Additionally, it uses dragging and multi-touch gestures to take further advantage of the expressiveness and playability offered by touch screens [8].

The concept of crowdsourcing musical creation through mobile technology has been explored previously, perhaps starting in 2001 with *Dialtones - A Telesymphony* by Golan Levin[10], where phones in the audience were dialed by the performers using custom control software that allowed up to 60 phones to be dialed simultaneously. Moving from local to networked, *World Stage* has been explored in both research and products by Smule, a mobile software developer

focused on social music applications. World Stage offers a place for a community to score arrangements for each other, perform music to one another, and even anonymously judge performances[15], all on a mobile device.

3. DESIGN EXPLORATION

3.1 The Interactive Album

The initial intent of this research was to create an interactive album. The goal was to give the user a sense of interactivity that, unlike a mash-up or remix, did not overtake the composer's structure. Another objective was to compose and experience interactive music written specifically for the "popular music" domain, meaning that it would be accessible to a large audience outside of the realm of computer music. For this to work, the experience would need to be powerful no matter how much the listener chose to interact with it, creating a unique but always desirable output with every listen. Initial research left numerous questions ranging from psychology to system design: *What does it mean to write a "hook" in a non-linear, event-triggered soundscape? What is the balance between the control one maintains as a composer versus as a listener? What interactions can one leverage from available devices to manipulate a composition?*

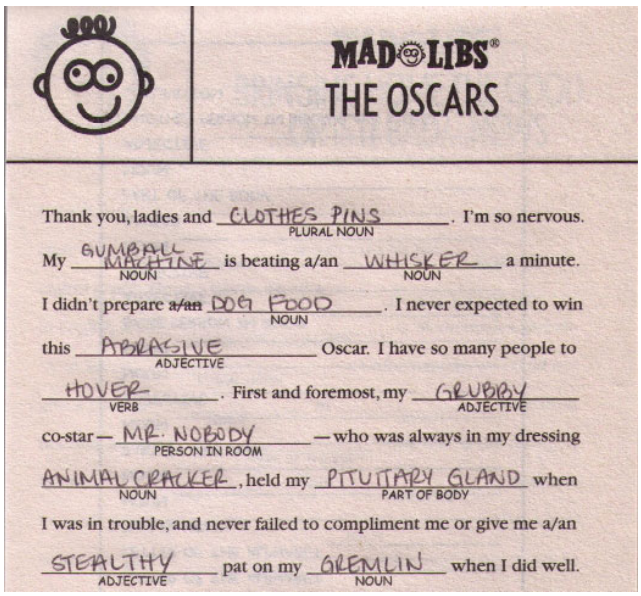


Figure 5: An Excerpt from a completed original Mad Libs. This page is only revealed *after* the words have been chosen.

3.2 From Mad to Madder

The first stab at addressing these questions was entitled *Madder Libs*. It was designed through the metaphor *Mad Libs for audio*[1]. The basic premise of Mad Libs (Figure 5) is that the player is given a page with several blanks to fill in, and a basic category for a word that he or she will choose to fill in each blank. Although it is known that these choices will fill in key words for a small story, nothing about the structure or content of that story is revealed, so the player must choose the words almost blindly, based on the given hints [11]. *Madder* Libs is very similar to this. A composer creates a song that does not produce any sound, but is a musical blueprint that pictorially hints at what sounds are to be used for each note (Figure 6). It is the listener's responsibility to record a sonic interpretation of the picture for each note without knowing the structure or content of the song, and upon completion the user can hear the song played with these new personalized sounds. In this way, the structure of the composition is maintained while still allowing the user to have a novel and personalized experience. At its core, Madder Libs is a non-traditional notation system that utilizes audio technology to make quick recordings rather than have the sound for each note repeatedly performed live. An extension that this technology offers is the ability to manipulate and replay these clips with accuracy, repetition, and speed beyond the limits of human ability, allowing the composer to write, for instance, extremely fast or lengthy passages without needing to worry about the limitations of the performer.

In response to the numerous questions raised about interactive album making in the previous section of this paper, this single interaction was in no way a complete answer. However, the insight gained proved valuable, and provided the foundation and the etymology for MadPad.

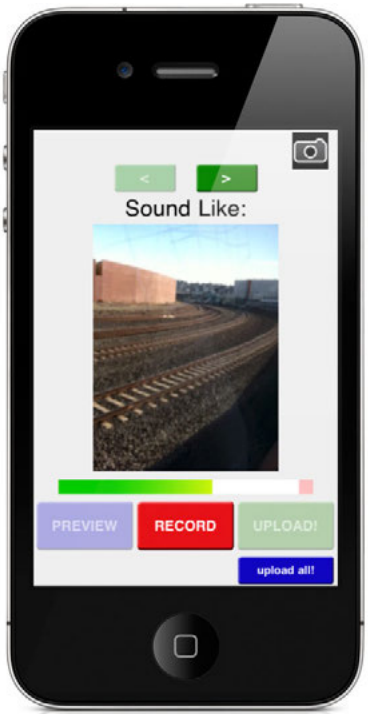


Figure 6: The Madder Libs recording interface, a predecessor to MadPad.

3.3 Bringing In the Network

At an early stage of the Madder Libs design process, it was entered into the program for The Stanford Mobile Phone Orchestra's[14] Fall concert, which was themed around audience participation. With the possibility of many audience members recording sets of audio samples for the same composition, the new goal was to amass these sets quickly to a single location, at which point they could be called back down to the audio player and added to the song. This was achieved by creating a networked database to which all sound clips for a song were submitted. The database could then be queried, and the desired samples could be downloaded and dropped into the song at any point. The resulting performance was no longer solely a personal experience, but rather it was one shared by the audience, whose members contributed the content. To ensure that

[1]Hopefully it is easy to see why one might consider this to be Madder.

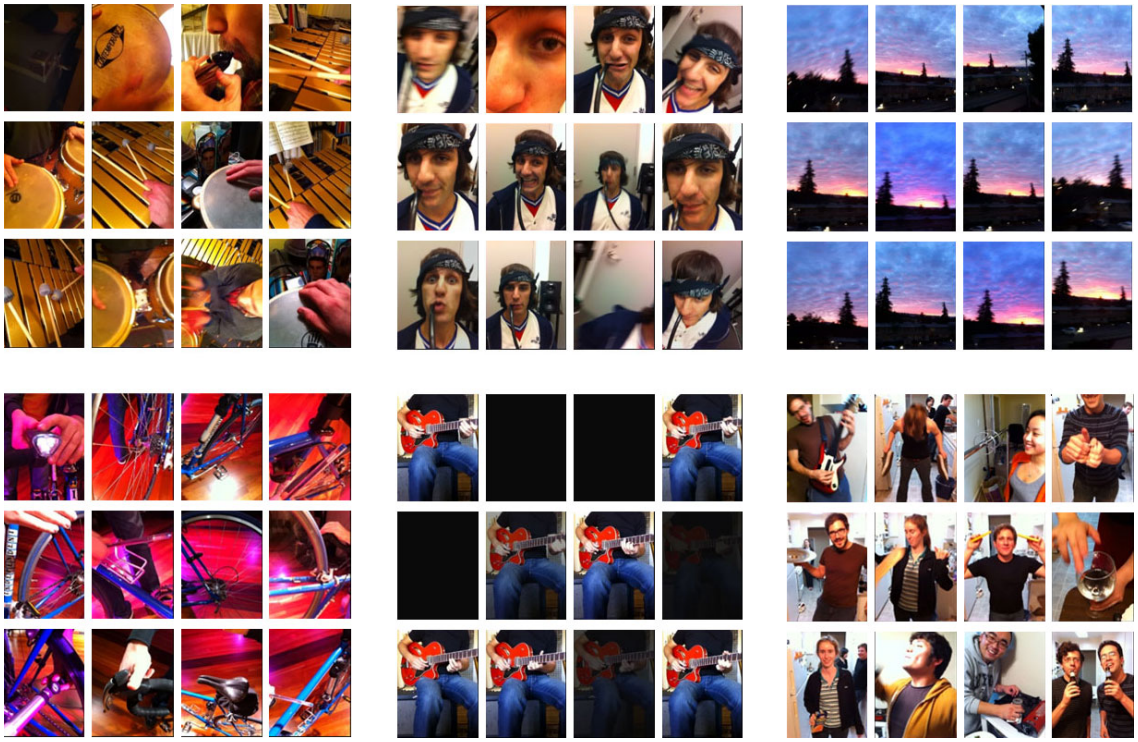


Figure 7: Six screenshots of user generated MadPads. Top left: Sampling a band. Top Center: Sampling a person playing a talkbox. Top Right: Capturing a sunset. Bottom Left: Turning a bike into an instrument. Bottom Center: Sampling a guitar (Some images have are faded because "Ghost Mode" playback is enabled, which fades away videos when they have stopped playing) Bottom Right: Remembering a dinner party.

all participants were given a chance to be included and to make each performance unique, samples were programmed to swap throughout the song. The concept of crowdsourcing content and making everyone feel like a part of the performance would become an important feature of MadPad, extending this notion from a local crowd to a global system.

3.4 From Madder Libs to MadPad

As preparations for the concert continued, we decided to add a visual component to the Mad Libs metaphor. Not only would we record a sample of the participant's voice, we'd also record the corresponding video–acquiring plenty of fodder for our projector, but more importantly giving the audience a way to connect the sounds they were hearing with the people who performed them. This emergence of the "What you see is what you hear" concept would become a main pillar of the MadPad experience. At this point the samples were laid out in a grid pattern[2] on a computer screen and triggered by precomposed MIDI messages, but it wasn't long after seeing this arrangement that the desire grew to trigger those clips on the fly and on-the-go. The whole system was ported to the iPad, utilizing both the large, multitouch surface and brilliant color display, as well as making the cloud-based social aspects of the system mobile. The concert was performed successfully on the iPad as a combination of precomposed MIDI, live performance on the device, and random audience-sourced sample swapping.[3]

[2]Originally there were plans to offer a wider array of layouts and transitions and this is still being considered as a future implementation.

[3]After the performance, the ability to play precomposed MIDI was removed from the feature set, as it was considered a potential source of confusion to the average user. Future work intends to include abilities to sequence and record shareable performances.

4. CORE FEATURES

4.1 Adding Video To The Mix

The primary interaction of MadPad[4] employs the extension of sample-based, tap-triggered music to include both audio and video. As a general concept, triggering video samples on-the-fly existed before MadPad, but not for multitouch devices. Tapping in to this new interface is what separates MadPad from its audiovisual sampling predecessors. For one, the recent ubiquity [3] of multitouch devices makes the interaction much more accessible to everyday people, and reaching a large audience has always been a primary goal. Additionally, multitouch screens give performers the ability to control the videos under their fingertips, as if the pads of an MPC were replaced with individual video screens. One can infer the interaction almost instantly–touch a picture to make it play. The system itself is intended to be a generic platform[6], and recedes into the background, allowing the content to shine through and encouraging a natural sense of wonder and exploration.

4.2 Make An Instrument Out Of Anything!

When using the MadPad to create content, video clips of anything can be loaded into the sample slots, and the possibilities are only bound by the user's surroundings and imagination. This finds shared ground with the concept of musique concrète, wherein *(translated from French) "The compositional material is not restricted to the inclusion of sounds derived from musical instruments or voices, nor to*

[4]"MadPad" as a name was initially just a joke. The project filename was the hasty concatenation of "Madder Libs" and "iPad" when we were just doing an initial test to see if the iPad could even load this amount of data to memory. Naturally, the name stuck.

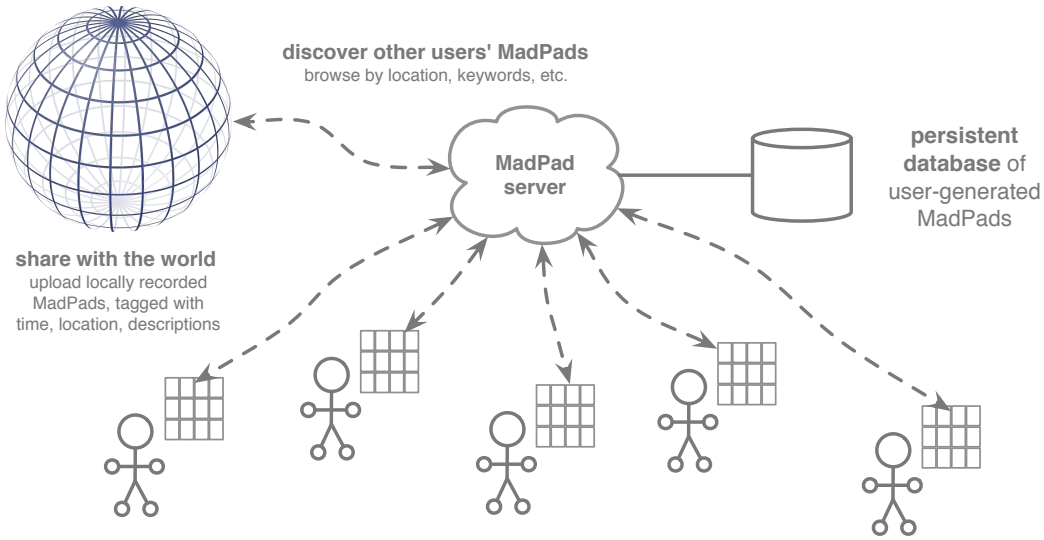


Figure 8: MadPad Community: Users serve as both content producers and consumers.

elements traditionally thought of as 'musical' (melody, harmony, rhythm, metre and so on)."[5]. However, whereas the notion of acousmatic sound in musique concrète is to intentionally separate the sound from its origin[2], MadPad's philosophy is quite the opposite. Although the concept of "found sound" is central to both musique concrete and MadPad, by its very nature MadPad encourages the association of the "found video" as well. Rather than concealing the sound source, MadPad exhibits the antithesis: What You See Is What You Hear (WYSIWYH)[5]. Interestingly, one can choose to deliberately rebel against this aesthetic and create samples where the audio and the visual have no direct correlation (e.g., a video of a marimba being struck while hearing the sound of a duck).

The WYSIWYH concept provides an additional level of personal connection to traditional sampling. By elucidating the sound source with highly contextual visuals, the system aims to generate a more holistic and immersive experience. It transforms the act of *sampling* into a type of *instrument design*. By allowing this interaction, WYSIWYH on MadPad attempts to open up the minds of its users to view everything they interact with as a potential source of music–perhaps its the sounds on the bus or the footsteps of different shoes. In addition to recording objects not intended to be musical, one can sample notes from an actual musical instrument and play them in a different way. (See figure 7 for more examples.) The MadPad platform thrives on the creation of unique, personalized musical instruments from anything important or interesting in the lives of each individual user.

4.3 A Social Sampler

Creating an environment to share content is another important feature of MadPad. Although standard sample sets[6] are readily accessible, fresh user-generated content is available through a MadPad Community. (Figure 8). The concept of social music content generation is explored in the design of Smule's Ocarina, an iPhone application that transforms the phone into an expressive, flute-like instrument[13]. The Ocarina community uses a simple tablature notation system to share popular melodies with its users in the form of an online songbook, with over 2,000 songs currently viewable. Thus the value of the Ocarina is constantly being enhanced due to the dedication of the user base. Similarly, in MadPad the social aspect autonomously extends the available content. In addition to creating samples, the user can browse clips from users around the world. The community serves as a forum for sharing creative ideas and collaboratively developing new ways in which the MadPad platform can be used for musical expression. This adds value for the viewer of the content, and it also adds a new level of drive for the creator. Knowing that one's concept of a musical instrument will be viewed by anonymous people around the world can motivate the production of more content and the innovation of more ideas for what these twelve empty slots can do, continuing to enrich the community.

Samples can also be discovered based on location. They are loaded as a conglomeration of the twelve closest samples made by distinct users. This mode allows a user to load in a set of samples recorded in a chosen geographical region and play an audiovisual instrument based on the collaboration of people who might be complete strangers to each other, but all share a similar proximity (e.g. loading in twelve samples from twelve different users in downtown Chicago).

In addition to offering anonymous sharing, MadPad also offers an ability to share locally without a network. For instance, friends at a party can take samples throughout the night just as one might snap photos. The result is an instrument that documents small snippets of the events that transpired. This type of scene capturing is a novel form of persistent media that bridges the gap between a photo album and a video, in that it offers a quickly digestible and *interactive* way to relive the moment. In another example, many people can pass around the camera and take turns recording the samples which will ultimately become a final instrument. This collaboration allows each performer to give individual input into what the instrument should be, and the result is a unique mix of personality and imagination representative of the group (Figure 7, upper left, is an example from a social gathering). After performing *for* the recording, the group can immediately gather around and continue to perform *with* the recording, closing the loop of the MadPad system.

[5]And commutatively, What You Hear Is What You See (WYHIWYS)

[6]Standard instrument sets and quirky idea sets are permanent, downloadable links, bundled with the application when downloaded.

5. CONCLUSIONS

MadPad began with the desire to create interactive compositions and evolved into a social/mobile platform for audiovisual creativity and collaboration. The ability to tap on a picture and make it play serves to bring creativity out of those who are not familiar with a traditional audio sampler, while also giving those who are familiar with it a new dimension to their creativity. Using the platform to create an instrument out of anything one sees encourages people to view the world as a more musical place. Giving people a place to share their creations allows them to learn from each other, and see the musical world that exists in every person's life–that is always present.

6. ACKNOWLEDGMENTS

This research has been generously supported through the Carmen Christensen Fellowship Award as well as the National Science Foundation Creative IT grant No. IIS-0855758. We would also like to thank David Kerr for his editing assistance and for first suggesting to move the system to the iPad.

7. REFERENCES

[1] Akai Pro. `http://www.akaipro.com/mpc`. Retrieved January 2011.

[2] M. Chion. *Audio-Vision Sound on Screen*. Columbia University Press, July 1994.

[3] J. Colegrove. The state of the touch screen market in 2010. DisplaySearch Touch Panel Market Analysis. Retrieved 2011-01-25.

[4] J. Conte. VideoSong 1 - Push - Jack Conte. Online video clip. YouTube, March 2008. Retrieved 2011-01-25 from `http://www.youtube.com/watch?v=FUVgPjnEMzw`.

[5] J. Dack. Technology and the instrument. *musik netz werke - Konturen der neuen Musikkultu*, 2002.

[6] G. Essl, G. Wang, and M. Rohs. Developments and Challenges turning Mobile Phones into Generic Music Performance Platforms. In *Proceedings of the Mobile Music Workshop*, Vienna, Austria, 2008.

[7] L. Gaye, L. E. Holmquist, F. Behrendt, and A. Tanaka. Mobile music technology: Report on an emerging community. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 22–25, Paris, France.

[8] G. Geiger. Using the Touch Screen as a Controller for Portable Computer Music Instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Paris, France, 2006.

[9] L. Gjertsen. Amateur. Online video clip. YouTube, November 2006. Retrieved 2011-04-25 from `http://www.youtube.com/watch?v=JzqumbhfxRo`.

[10] G. Levin. Dialtones - a telesymphony, September 2001. Retrieved 2011-04-25 from `http://www.flong.com/projects/telesymphony/`.

[11] Penguin Group USA. Mad libs. Retrieved 2011-01-25 from `http://www.madlibs.com`.

[12] Two Hand Band. History and Significance of the MPC. Documentary Film, August 2008.

[13] G. Wang. Designing Smule's iPhone Ocarina. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Pittsburgh, USA, 2009.

[14] G. Wang, G. Essl, and H. Penttinen. Do Mobile Phones Dream of Electric Orchestras? In *Proceedings of the International Computer Music Conference*, Belfast, 2008.

[15] G. Wang, J. Oh, S. Salazar, and R. Hamilton. World Stage: A Crowdsourcing Paradigm for Social / Mobile Music. In *Proceedings of the International Computer Music Conference (under review)*, Huddersfield, UK, 2011.

[16] L. Werthheimer. Pomplamoose: Making A Living On YouTube, April 2010. National Public Radio.

The Visual in Mobile Music Performance

Patrick O'Keefe
University of Michigan
Electrical Engineering: Systems
1301 Beal Avenue
Ann Arbor, Michigan 48109-2121
pokeefe@umich.edu

Georg Essl
University of Michigan
Electrical Engineering &
Computer Science and Music
2260 Hayward Street
Ann Arbor, Michigan 48109-2121
gessl@eecs.umich.edu

ABSTRACT

Visual information integration in mobile music performance is an area that has not been thoroughly explored and current applications are often individually designed. From camera input to flexible output rendering, we discuss visual performance support in the context of urMus, a meta-environment for mobile interaction and performance development. The use of cameras, a set of image primitives, interactive visual content, projectors, and camera flashes can lead to visually intriguing performance possibilities.

Keywords

Mobile performance, visual interaction, camera phone, mobile collaboration

1. INTRODUCTION

Although mobile device interaction is tremendously visual, they inherently suffer from a limitation on screen real estate. However, this restriction is mitigated by the growing popularity of tablet devices and portable projectors; there are even some mobile phones on the market with integrated pico projectors. This indicates a general consumer interest in transcending these visual limitations and making the mobile experience more communal.

The purpose of this paper is to make the visual modality an accessible part of mobile music performance. This includes both the built-in cameras as sensor input as well as the screen and projected images as output. When incorporated into a flexible graphics and data-flow engine, it becomes possible to rapidly develop performances that seamlessly integrate computer vision, sound synthesis, and rich visual output. With the use of many mobile devices with projectors, visual display becomes more modular and a coordinated effort. The relative position of performers and the choice of projectable space expand what mobile performances are even conceivable.

For our implementation of these visual concepts, we have worked in the context of urMus, a meta-environment of mobile device programming for artistic purposes [7, 9, 8]. The goal of urMus is to make the design of all aspects of mobile phone interactions and performances easy and flexible at the same time.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. RELATED WORK

The use of the camera for mobile phone interaction has been explored extensively, but not in a musical context [21, 20, 10, 19, 24, 18, 17]. There have also been non-mobile studies on mapping computer vision features to sound [13, 3]. In a musical context, phone cameras have been used for motion detection when mobile phones at the time did not have built-in accelerometers or other motion sensors [22, 23]. In one paper, the mobile phone was played as a wind instrument using the microphone as the wind sensor. Covering the camera (detectable by overall brightness) would act as closing a tone-hole in the wind instrument [12]. A predecessor to urMus, SpeedDial, was a Symbian mobile synthesis mapping environment which used the camera as an abstracted sensor and allowed overall camera brightness to be mapped to control a range of synthesis algorithms [6]. Mobile music making itself is an ongoing topic of interest [28, 2]. Specifically, the importance of the visual has been recognized and explored in the work of the Mobile Phone Orchestra (MoPho) [26] in the design of Ocarina [25] and Leaf Trombone [27] as well as in the visual layouting support of urMus [9].

There have also been several studies into the new types of interaction and experiences provided by coupling a portable projector with mobile devices [29, 15, 1]. Work by Cao has explored multi-user portable projector interaction and different types of projectable spaces [5, 4].

3. VISUAL INPUT

Digital cameras on contemporary mobile devices have high image quality and offer very fast rates of capture. One can interpret the information provided by the camera as literal – images that represent a world are to be interpreted and displayed as presented – or this information can be abstracted and used to drive performance. The goal of this work is to explore both options.

In order to enable each interpretation, we give access to camera information though two possible routes. One method is a part of a data stream pipeline where camera images are reduced to single numbers which in turn can be used to control sound synthesis. Rather than using detailed visual information, broad features of the camera image are used to provide control. The second method is access to the full camera image itself. This data is accessible via a rich OpenGLES-based rendering system that can be used to create new and diverse visual content. In this section, we will discuss how the information from the camera is received from the operating system and processed.

3.1 Access to Video Data

For iOS devices, official APIs to access video data were made available with the iOS 4.0 software update in the AV-Foundation Framework. Since our current implementation

Image Neighborhood

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Roberts Edge Detector Masks

-1	0
0	1

$g_x = z_9 - z_5$

0	-1
1	0

$g_y = z_8 - z_6$

Figure 1: The Roberts Edge Detector masks and the first order derivatives they approximate.

is only for these devices, this is what will be discussed here. Upon application launch, an AVCaptureSession is created for the rear-facing camera with a request to process fifteen frames per second. Most importantly, the AVCaptureSession is configured to process these frames asynchronously on a secondary dispatch queue. This ensures that the user interface and other signal processing tasks (such as audio output) are not interrupted. Moreover, the secondary dispatch queue drops late video frames when the system cannot handle the requested fifteen frames per second. In practice, this happens quite often but is completely transparent to the user.

There are also three configurable aspects of the video capture process. The first is camera selection. Most iOS devices only have a rear-facing camera, but the iPhone 4 and fourth-generation iPod Touch have an option to select the front-facing camera as well. Currently, it is not possible to get data from both sources at once. The other two aspects allow the choice between an automatic or fixed setting for both the white balance and exposure. This has interesting implications for certain low-level visual features. For example, if overall brightness was being used to drive the frequency of a sine oscillator, a fixed white balance and exposure would be necessary to achieve a low frequency when the camera was covered and a high frequency when pointed at a light source. However, automatic white balance and exposure settings would result in the sonification of these processes – something that could be desirable.

3.2 Visual Features

There is no standard set of visual features that are applicable to performance situations. In the context of the urMus environment, features need to be expressed as a floating point value (or array of values) between negative one and one. To maintain generality and for computational considerations, the features developed are low-level in nature.

The first four features are overall brightness, red sum, blue sum, and green sum. Their computation is nearly self-explanatory. For a pixel buffer with n rows and m columns, the overall brightness is computed as follows.

$$\text{brightness} = \frac{1}{3nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(\text{red}(i,j) + \text{green}(i,j) + \text{blue}(i,j))$$

The image processing community has developed many ways to quantify color in an image. Sometimes this involves different color spaces, color independent of brightness, and the biological processes behind color perception. Interesting situations arise: should a bright white wall have a higher "red" feature than a rose petal? For simplicity, we simply sum the components in the respective RGB channels and divide by the maximum. As mentioned above, having a locked white balance and exposure for these four features is most likely desirable.

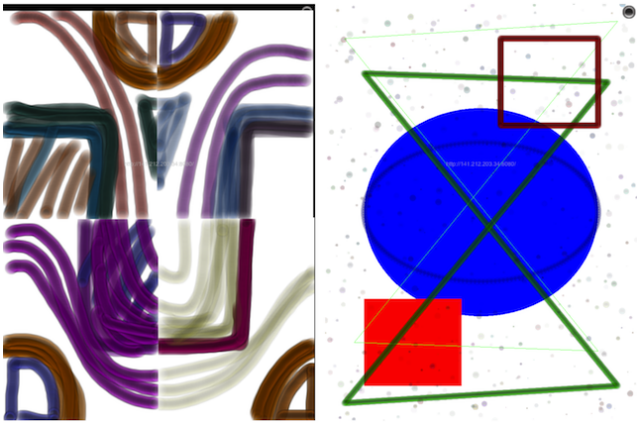


Figure 2: Examples of 2D rendering in urMus: A tiled canvas drawing program (left). The text image showing the standard drawing elements (right).

Another feature is simply named "edginess." To compute this feature, we use the standard *total variation* metric given by

$$T(y) = \sum_{i,j}\sqrt{(y_{i+1,j} - y_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2}$$

For our implementation, the Roberts Edge Detector is used to approximate the first derivates between adjacent pixels. Figure 1 shows the masks and corresponding first order derivative approximations for this detector. The Roberts detector is one of the oldest edge detectors and is frequently used in embedded applications where simplicity and computational speed are paramount [11]. The final "edgy" feature is normalized to fall between zero and one.

Other low level image features can be easily incorporated into urMus at this stage. Certain features, such as optical flow, have already been investigated for mobile devices and would be natural for inclusion [22]. Higher level features, such as the x and y coordinates of a detected face, could also be considered. However, these more complex features have a much higher computational complexity and would greatly reduce the rate at which images are processed. Since all of the features are calculated for each camera frame received, the feature with the highest complexity will be the limiting factor.

4. VISUAL OUTPUT

One goal of urMus is to provide an environment in which rich interactive media content can be written and designed. Part of that goal includes trying to find the right kind of programming language abstraction to make visual programming immediate and easy, yet as flexible as possible. This is in the spirit of visual programming and code as art as embodied in the design of Processing [16]. OpenFrameworks, a set of libraries in C++, has also been ported and used with iOS devices and is quite popular for mobile art projects [14]. The goal is to be much closer to the concept of Processing

Figure 3: Examples of live camera feeds within multiple regions in urMus subject to a range of texture and color transformations.

and other specialized environments for art programming by gearing not only the API but also the environment and language for the task at hand.

urMus already comes with a rich and flexible graphical layout system that uses the concept of textures to create visually appealing details [9]. In order to allow visual programming, a texture in urMus acquires two functions. The first is that of a canvas. Graphical manipulation primitives can be applied to a texture to render into it. Currently urMus supports a set of graphical drawing primitives that is close to the set offered by Processing for 2D rendering. For the second function, textures also serve as brushes that can be used with any rendering primitive. This makes it easy to generate fairly complex visual content with simple primitives thanks to the use of complex and possibly changing textures. Furthermore, one can explore iterative and recursive painting ideas by repeatedly changing the texture roles of brush and canvas. Finally, as each texture can be flexibly moved, resized, and rotated on screen via the layout engine, one has versatile interactive control.

The camera image should be a flexible component of a visual mobile part piece. Current solutions often are inflexible. In most mobile situations, the camera input is just directly mapped to the full size of the screen. In urMus the camera image is directly fed into an OpenGL texture, which can be used in arbitrary number of instances and independently manipulated. This has a number of implications. For one all the standard texture and region manipulation capabilities of urMus do apply, such as tiling, rotating, stretching, and skewing that is possible by changes of texture coordinates and size of the containing region. Furthermore the camera texture can also be used as brush, hence one can actively draw and use all the 2D drawing primitives discussed earlier in this paper. This flexibility gives the artist many interesting ways to display what the camera "sees." At the same time the camera becomes part of the repertoire of visual information to create new content. These uses of the camera are fully interactive and multiple instances of camera images can be manipulated independently.

4.1 Rendering Primitives

The 2D rendering primitives of urMus can be roughly categorized into three groups. The first group consists of actual drawing functions which are `Point(x,y)`, `Line(x1,y1,x2,y2)`, `Rect(x,y,w,h)`, `Quad(x1,y1,x2,y2,x3,y3,x4,y4)` and `Ellipse(x,y,w,h)`. These allow for the display of points, lines, rectangles, arbitrary quadrangles, and ellipses. The second set of functions influence how these primitives are rendered: `SetBrushColor(r,g,b,a)` sets the brush color, `SetBrushSize(s)` changes with width of the brush, and `SetFill(b)` toggles whether or not the primitive is filled (if it is a closed primitive such as a rectangle or ellipse). The last set is texture control. If the command `UseAsBrush()` is invoked on a texture then future drawing and brush commands will use this texture as brush. This will continue until another texture is assigned as brush. All these operations are member functions of a texture, hence any texture can be drawn into, and any texture can be assigned to be the current brush. Finally, as any texture is by definition part of a region in urMus, it can be flexibly resized, positioned, layered and tiled.

Figure 4: An ensemble setup of mobile projectors and their driving mobile devices.

Figure 2 shows the results of examples written in urMus. The leftmost example shows a tiled canvas drawing program. The canvas consists of four independent regions, which can be locked, unlocked and moved around on the screen. The painting will take the changed position into account, leading to the ability to continuously reiterate over the same image with different canvas arrangements, creating changing symmetries. The right example is the generic 2D rendering primitives text showing line, filled, and texture-based drawing of all basic primitives available. The thick-lined primitives are using a circular texture as brush and alpha-blending is active.

4.2 Output Technologies

Our work has looked at three different ways to extend the visual output capabilities of a mobile device for performance situations. The mobile multi-touch screen itself already serves as a rich and very useful display and larger portable devices such as the iPad extend the visual possibilities. Yet advances in mobile projector technologies are allowing to further expand and change the types of visual display that are possible. This technology is still in its infancy but is already quite useful. We use Aaxa Technologies pico projectors which offer 33 lumens of intensity at a battery-time of roughly 30 minutes. This is too low for use in an ordinary lit room, but quite useful in rooms with dimmed lighting conditions. With this technology it becomes possible to tile multiple images, project on arbitrary surfaces and objects and create varied visual content while on the move (see Figure 5). The projectors can serve as much as a flash-light as a display. The mobile projectors are connected to the device using a video-to-dock connector and OpenGL content can be rendered onto external displays at interactive rates. Currently we are using ten such projectors (six of which can be seen in Figure 4).

The last form of output we have considered is the camera

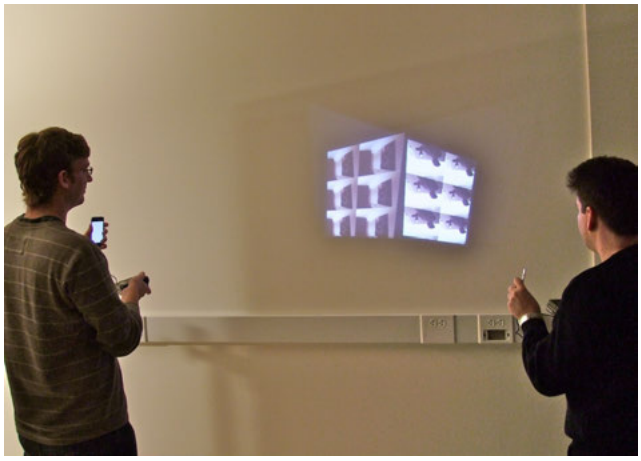


Figure 5: Two devices with pico projectors displaying a composite image based on their respective camera inputs.

flash. The iPhone 4 contains a powerful LED flash and the feature is becoming standard on the latest mobile devices. The flash can be set to turn on and off at a variable rate which creates a stroboscopic effect.

4.3 Camera Integration

In the context of urMus, the graphical display is entirely controlled by OpenGL which has the benefit of being cross-platform. Also, it instantly gives the graphical versatility we desire. Once the camera input has been rendered to an OpenGL texture, any kind of transformation can be applied without effecting other instances. As mentioned above, the camera images are processed asynchronously on a secondary thread which is necessary to keep the user interface responsive. This presents a problem for the OpenGL pipeline because only one OpenGL context can exist on a thread at a time. To work around this, a new context is created on the secondary thread that uses the same sharegroup as the main thread's context. When two contexts are members of the same sharegroup, all texture, buffer, and framebuffer object resources are shared. When the very first frame of camera pixel data is received, a texture is created, the pixel data is copied into the texture with `glTexImage2D()`, and the main thread's context is made aware that it has access to a camera texture. All subsequent camera frames render into the texture using `glTexSubImage2D()` which redefines a contiguous subregion of an existing two-dimensional texture image. This eliminates the need to re-create textures with every new frame which saves computational costs and also prevents interference between the actual display of the texture on the main thread and the texture update process. Following this approach it is possible to retain interactive rates even if multiple copies of the camera texture are in use.

Access to camera texture is made possible through an extension of the texture API of urMus. By setting the `UseCamera` option of a texture instance, this texture will start using the current camera texture for all its texture-based operations. If the device offers multiple cameras (such as a front and a back-facing camera), these can be selected using the global `SetActiveCamera(cam)` API function. Currently all active camera textures are affected by this, as it current iOS devices do not allow multiple cameras to be active at the same time.

Currently iOS cameras operate at 30 frames per second and we found that multiple camera textures can be active

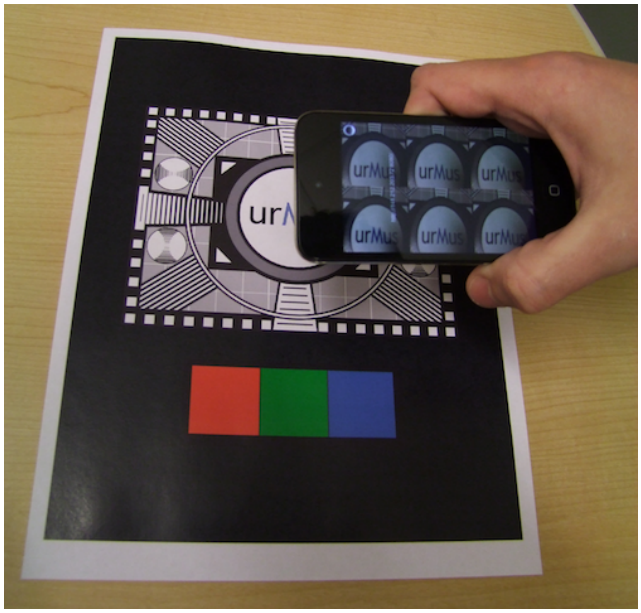


Figure 6: Features of the camera image are used to control audio.

while retaining interactive and that the performance is independent of the choice of camera. A test case with 30 active camera textures of various sizes gave a performance of 25 frames per second display update on an iPod Touch. The lamp of the flash can be made to oscillate at a fixed rate using the `SetTorchFlashFrequency(freq)` global API.

4.4 External Display Integration

Since iOS 3.2 it is possible to be informed about an external display being plugged in and its resolution. One can then attach views that will be rendered on the external display. Currently in urMus, an external screen is automatically detected and the OpenGL rendering is redirected to the external display.

A test image with six live camera textures will render on an external display at 30 frames per second or above using an iPod Touch. This frame rate varies by less than 5 fps if the resolution of the external display is changed.

5. EXAMPLES

A vast area of performances can be imagined using the techniques discussed above, ranging from the use of the mobile device's display as visual augmentation to complex uses of camera input coupled with multi-media outcomes. Next we discuss a few possible examples that we have implemented so far using urMus.

5.1 Performing the Image

"Performing the Image" is a visual performance that uses a prepared printed sheet with with color and textures to allow performance of sound over the image. Using the live-patching graphical interface of urMus, the performer can change the sonic realization of the image on the fly with simple multi-touch interactions by using features extracted from the camera signal as sources to drive synthesis patches. Color and edgy aspects of the camera image create a performantive canvas which can be explored by moving over different regions of the sheet (see Figure 6). This gives the piece a synesthetic quality by transforming the visual into the sonic.

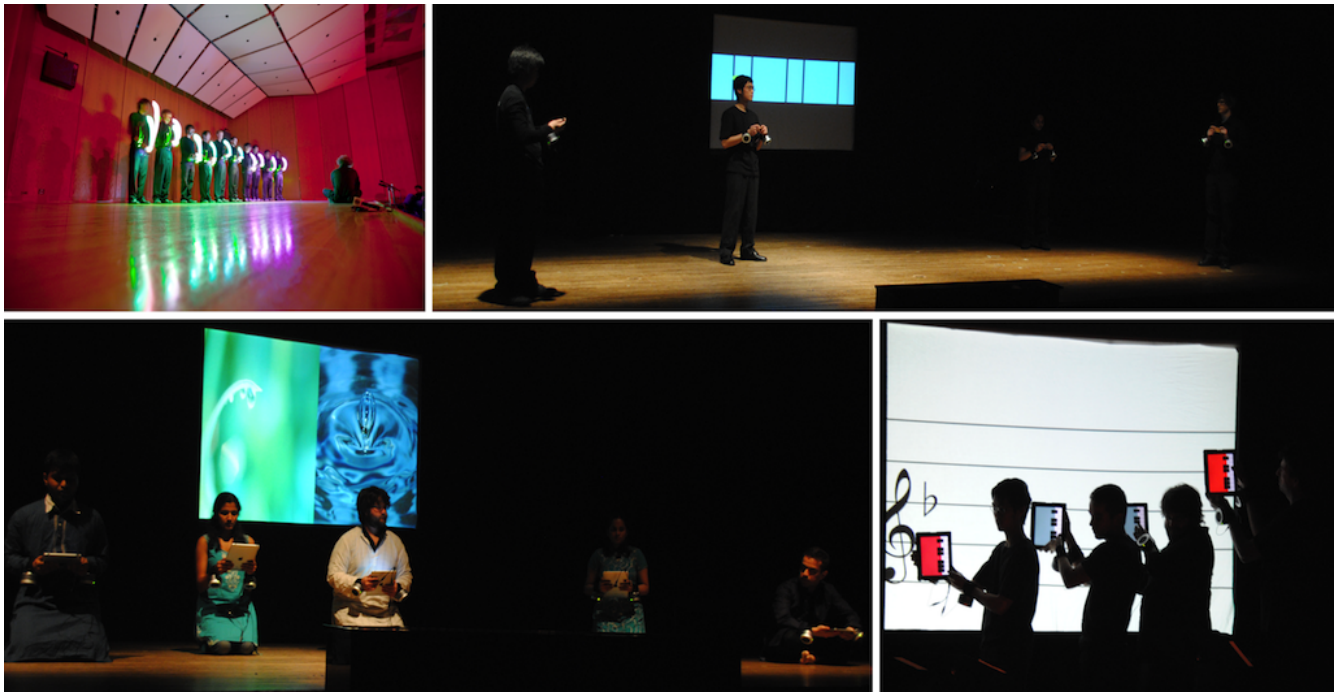


Figure 7: Examples of the Visual in pieces of the Michigan Mobile Phone Ensemble.

5.2 Visual in Mobile Phone Ensemble Performances

A key problem in mobile music performance is the explanation of the performance to the audience. There is no canonical understanding of what mobile music performance should be and very often visual communication is a big part of this explanatory task. A good example of this is the piece Color Organ, written by Kyle Kramer as part of a class taught at the University of Michigan on the topic of using urMus. As seen in the bottom right of Figure 7, four performers stand in front of a back-projected screen showing a musical staff. The performers hold the mobile device facing the audience. The screen of the mobile device itself is critical for explaining the piece to the audience. The performers lift the devices and place them in the correct position on the staff while colors express octave-matched notes.

Even static information can help strengthen the perception of a piece to the audience. The piece JalGalBandi by Guerrero, Dattatrhi, Balasubramanian, and Jagadeesh uses visual projection to reinforce the sound. The piece transforms traditional Indian performance into an ecological perception of water and the visual display helps reinforce the kinds of water sounds that are currently creating the sonic experience (see bottom left of Figure 7).

Space Pong by Gayathri Balasubramanian and Lubin Tan uses networked communication to pass a virtual ball between performers. While gestures to symbolize that a ball is being passed around, the networked communication of the piece is not apparent. After all the transitions of sounds could have been due to actions of each individual performer and not some exchange. Here the projected visual display is also included in the network and depicts the interactions and changes that are induced by the performer's actions and it creates a visual appearance of an virtual ball moving in a virtual performance plane (see top right of Figure 7).

The importance of visual communication becomes even more critical if the devices used are small. In the Ballad of Roy G. Biv by Devin Kerr the screens of mobile devices are turned into mobile colored dot arrays. The piece is performed completely in the dark. Figure 7 shows a long exposure shot of the performance. Each color has a musical loop associated with it and the change of gestures in space create phasing effects and interplay that is intricately linked with the visual appearance of the piece (see top left of Figure 7).

6. CONCLUSIONS

In this paper we discussed a range of aspects regarding the visual in mobile music performance. Visual information can be used both as input and as output in musical performance. In order to make it easy for artists to create new mobile music performances with visual contributions, we have discussed how both visual input and output is facilitated within urMus, a mobile performance meta-environment. By combining camera capture with the generic OpenGL texture rendering engine, camera images are made flexible and objects of manipulation. Combined with textural rendering primitives, the camera can become a brush. For output, we discussed how textures can serve as both canvas and brush and therefore lead to a range of visual performance ideas such as rearranging canvases or recursive visual content. The emergence of mobile projectors extends and liberates the visual display, and multiple performers can join in creating content not just by what is shown, but also by where it is directed or moved.

Current technology is still limited by the computational power of the mobile device. While simple computer vision algorithms can be easily implemented, richer visual features are still too expensive to extract at interactive rates. Finding ever more complex sets of visual control and display remains a topic for future work as does the exploration of the vast possibilities of mobile display technologies in interactive mobile performance.

7. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Student Research Fellowship under Grant No. DGE 0718128. This work was

supported in part by a Curriculum Innovation Grant by the Office of Undergraduate Affairs of the College of Engineering of the University of Michigan. Thanks to the student of the University of Michigan course "Mobile Phones as Musical Instruments": Gayathri Balasubramania, Yuan Yuan Chen, Chandrika Dattathri, Alejandro Guerrero, Andrew Hayhurst, Kiran Jagadeesh, Steve Joslin, Kyle Kramer, Billy Lau, Michael Musick, Lubin Tan, Edgar Watson.

8. REFERENCES

[1] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. Touch projector: mobile interaction through video. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 2287–2296. ACM, 2010.

[2] N. J. Bryan, J. Herrera, J. Oh, and G. Wang. Momu: A mobile music toolkit. In *Proceedings of the International Conference for New Interfaces for Musical Expression*, Sydney, Australia, 2010.

[3] A. Camurri, B. Mazzarino, and G. Volpe. Analysis of expressive gesture: The eyesweb expressive gesture processing library. *Gesture-based communication in human-computer interaction*, pages 469–470, 2004.

[4] X. Cao and R. Balakrishnan. Interacting with dynamically defined information spaces using a handheld projector and a pen. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 225–234. ACM, 2006.

[5] X. Cao, C. Forlines, and R. Balakrishnan. Multi-user interaction using handheld projectors. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 43–52. ACM, 2007.

[6] G. Essl. SpeedDial: Rapid and On-The-Fly Mapping of Mobile Phone Instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Pittsburgh, June 4-6 2009.

[7] G. Essl. UrMus – an environment for mobile instrument design and performance. In *Proceedings of the International Computer Music Conference (ICMC)*, Stony Brooks/New York, June 1-5 2010.

[8] G. Essl. UrSound – live patching of audio and multimedia using a multi-rate normed single-stream data-flow engine, 2010. Submitted to the International Computer Music Conference.

[9] G. Essl and A. Müller. Designing Mobile Musical Instruments and Environments with urMus. *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, pages 76–81, 2010.

[10] G. Essl and M. Rohs. Interactivity for Mobile Music Making. *Organised Sound*, 14(2):197–207, 2009.

[11] R. Gonzalez, R. Woods, and S. Eddins. *Digital image processing using MATLAB*, volume 624. Prentice Hall Upper Saddle River, NJ, 2004.

[12] A. Misra, G. Essl, and M. Rohs. Microphone as Sensor in Mobile Phone Performance. In *Proceedings of the 8th International Conference on New Interfaces for Musical Expression (NIME 2008)*, Genova, Italy, June 5-7 2008.

[13] K. C. Ng. Music via Motion: Transdomain Mapping of Motion and Sound for Interactive Performances. *Proceedings of the IEEE*, 92(4):645–655, Apr. 2004.

[14] OpenFrameworks. http://www.openframeworks.cc/.

[15] J. Park and M. Kim. Interactive display of image details using a camera-coupled mobile projector. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–16. IEEE, 2010.

[16] C. Reas, B. Fry, and J. Maeda. *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2007.

[17] M. Rohs. Real-world interaction with camera-phones. In *2nd International Symposium on Ubiquitous Computing Systems (UCS)*, pages 39–48, Tokyo, Japan, Nov. 2004.

[18] M. Rohs. Marker-based interaction techniques for camera-phones. In *IUI 2005 Workshop on Multi-User and Ubiquitous User Interfaces (MU3I)*, January 2005.

[19] M. Rohs. Visual code widgets for marker-based interaction. In *IWSAWC'05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems – Workshops (ICDCS 2005 Workshops)*, pages 506–513, Columbus, Ohio, USA, June 2005.

[20] M. Rohs. Marker-based embodied interaction for handheld augmented reality games. *Journal of Virtual Reality and Broadcasting*, 4(5), Mar. 2007.

[21] M. Rohs and G. Essl. Which one is better? – information navigation techniques for spatially aware handheld displays. In *ICMI '06: Proceedings of the 8th International Conference on Multimodal Interfaces*, pages 100–107, Nov. 2006.

[22] M. Rohs and G. Essl. Camus 2: optical flow and collaboration in camera phone music performance. In *Proceedings of the 7th international conference on New interfaces for musical expression*, pages 160–163. ACM, 2007.

[23] M. Rohs, G. Essl, and M. Roth. CaMus: Live Music Performance using Camera Phones and Visual Grid Tracking. In *Proceedings of the 6th International Conference on New Instruments for Musical Expression (NIME)*, pages 31–36, June 2006.

[24] M. Rohs and P. Zweifel. A conceptual framework for camera phone-based interaction techniques. In H. W. Gellersen, R. Want, and A. Schmidt, editors, *Pervasive Computing: Third International Conference, PERVASIVE 2005*, pages 171–189, Munich, Germany, May 2005. LNCS 3468, Springer.

[25] G. Wang. Designing Smule's iPhone Ocarina. In *Proceedings of International Conference on New Instruments for Music Expression (NIME)*, Pittsburgh, PA, 2009.

[26] G. Wang, G. Essl, and H. Penttinen. Do Mobile Phones Dream of Electric Orchestras? In *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, August 24-29 2008.

[27] G. Wang, G. Essl, J. Smith, S. Salazar, P. R. Cook, R. Hamilton, R. Fiebrink, J. Berger, D. Zhu, M. Ljungstrom, A. Berry, J. Wu, T. Kirk, E. Berger, and J. Segal. Smule = Sonic Media: An Intersection of the Mobile, Musical, and Social. In *Proceedings of the International Computer Music Conference*, Montreal, August 16-21 2009.

[28] G. Weinberg, A. Beck, and G. M. ZooZBeat: a Gesture-based Mobile Music Studio. In *Proceedings of International Conference on New Instruments for Music Expression (NIME)*, Pittsburgh, PA, 2009.

[29] M. Wilson, S. Robinson, D. Craggs, K. Brimble, and M. Jones. Pico-ing into the future of mobile projector phones. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, pages 3997–4002. ACM, 2010.

Designing for the iPad: Magic Fiddle

Ge Wang [§‡] Jieun Oh [§‡] Tom Lieber [‡]

Center for Computer Research in Music and Acoustics (CCRMA) [§]
Stanford University

Smule, Inc. [‡]
Palo Alto, CA, United States

{ge,jieun5}@ccrma.stanford.edu; tom@smule.com

ABSTRACT
This paper describes the origin, design, and implementation of Smule's *Magic Fiddle*, an expressive musical instrument for the iPad. Magic Fiddle takes advantage of the physical aspects of the device to integrate game-like and pedagogical elements. We describe the origin of Magic Fiddle, chronicle its design process, discuss its integrated music education system, and evaluate the overall experience.

Keywords
Magic Fiddle, iPad, physical interaction design, experiential design, music education.

1. INTRODUCTION
The father of ubiquitous computing Mark Weiser described a world where computing evolves from the "personal" to the "pervasive", where technology "disappears into the fabric of everyday life..." [16]. Weiser envisioned computing's evolution into "calm technology" that recedes into the background of our daily lives, empowering people without being noticed, "extending our unconsciousness". Today's personal mobile devices are becoming more powerful while our awareness of them as technology is shrinking.

The iPad, for example, engages users through a large multi-touch display and affords natural interactions that are more about "what to do" than "how". The iPad has potential to be incorporated into physical practice to the point that it is perceived by people as an extension of themselves.

The concept that people act *through* tangible artifacts, rather than *on* it, has been articulated by Klemmer *et al* and *Polanyi*, among others [7, 9]. We also share the sentiment expressed by Dourish that "tangible computing is of interest precisely because it is not purely physical. It is a physical realization of a symbolic reality" [4]; we hope to realize a tangible experience from our design of a musical instrument on the iPad by combining the physical (gesture and artifact) and the virtual (graphical interfaces and digital audio synthesis).

We are motivated by prior works that use mobile devices not simply as controllers or sensors, but as physical, tangible objects calling for meaningful gestural actions by their users [5]. One pioneering work that uses a commodity mobile device as physical musical instrument is *Pocket Gamelan* by Greg Schiemer [10]. In Schiemer's works, mobile phones are "mounted in a specially devised pouch attached to a cord and physically swung to produce audio chorusing."

NIME'11, 30 May–1 June 2011, Oslo, Norway.

Smule's *Ocarina* is another example that leverages iPhone's wide array of technologies – microphone input, multi-touch, accelerometer, real-time sound synthesis, and graphics – to create the physical experience of playing an ocarina. The Ocarina also creates a location-aware social experience by allowing its users to listen remotely to one another [13].

More recently, the Stanford Laptop Orchestra has explored possibilities in physical interaction techniques offered by mobile phones through their onboard sensors using the Mobile Music Toolkit [1]. For instance, Luke Dahl proposes a metaphor of "sound as a ball" to design *Sound Bounce* (2009), a gesture-controlled instrument that allows players to "bounce" sounds, "throw" them to other players, and compete in a game to "knock out" others' sound [3]. Another piece, *interV* (2009) by Jorge Herrera, uses the iPhone accelerometer to control sound using gestures such as gentle tilts and larger arm movements. *Wind Chimes* (2009) by Nicholas Bryan leverages mobile phones as directional controllers within a 8-channel surround sound audio system; the metaphor of wind chimes connects "physical" chimes (8-channel system) to a wind force (performer blowing air into the mobile phone's microphone input facing a particular direction) [8].

Figure 1. Playing a Magic Fiddle duet.

2. ORIGINS
The concept for Magic Fiddle began with a casual question: "can we design a violin-like instrument that is so *tangible* that users have to hold the iPad up to their face to play?" While seemingly absurd, this concept was attractive as a challenge to design such a physical interaction.

Thus we embarked on the research, design, and implementation of Magic Fiddle. The result was a unique and expressive musical instrument with game-like and pedagogical elements (Figure 1). The game aspects invite new users to start playing music with the instrument and over time, challenge users to engage a larger repertoire and strive for *virtuosity*. The pedagogical aspects present new and returning users with a fully-integrated, interactive music "teacher". The remainder of this paper presents and evaluates the process involved in creating the Magic Fiddle.

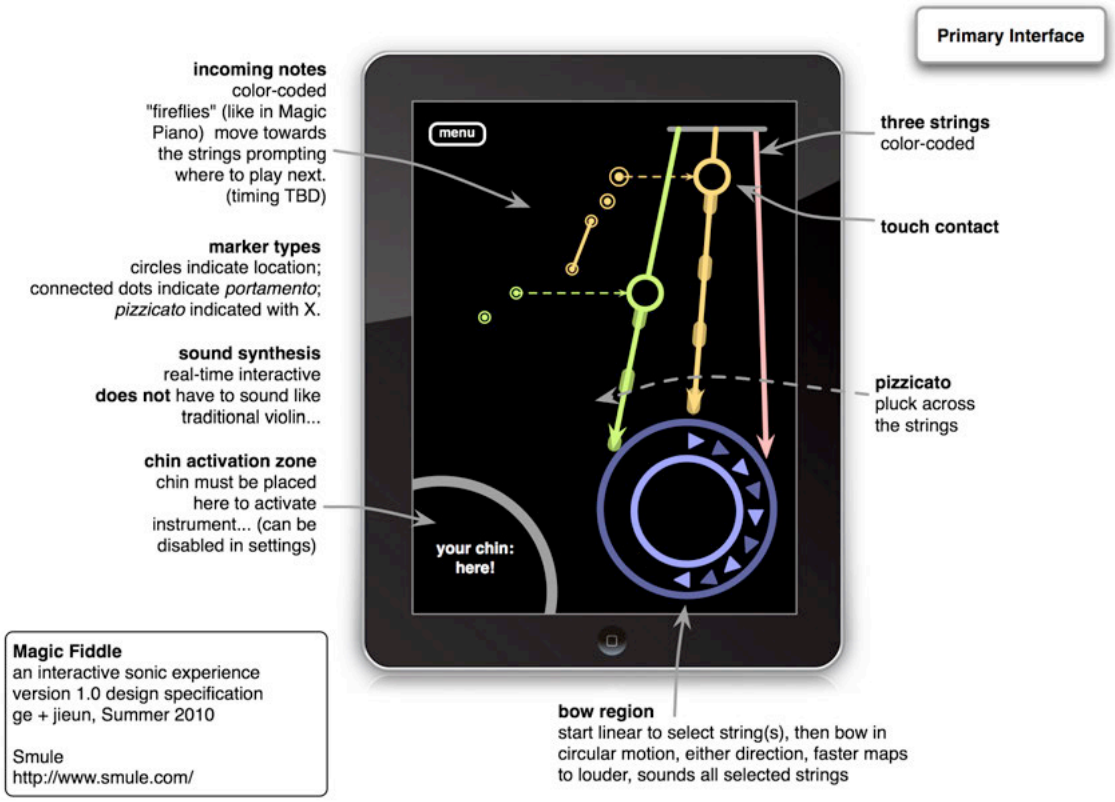


Figure 2. Preliminary Concept Design.

3. DESIGN PROCESS

Magic Fiddle was iteratively designed and implemented. We improved upon interface usability and musical expressivity by creating many prototypes. This section describes high-level issues we had to address and the implementation approaches we considered in response.

3.1 Fitting a Fiddle into an iPad

The foremost challenge we faced was re-purposing an iPad into something it was perhaps not meant to be. The iPad is physically smaller than a violin, such that "fitting" the instrument inside a screen having a diagonal length of 25cm is almost impossible. But even more critical than size is the device's affordance: a flat rectangular surface does not naturally suggest fingering and bowing as possible gestures.

We decided to use the violin as an inspiration, but not as an end goal. We began the design process acknowledging that an iPad cannot generate the same variety of realistic timbres as a physical violin, and cannot offer performers the fine-grained control required to reach comparable virtuosity in technique and expression. However, we believe that crafting a musical experience out of a personal mobile device is still "magical" in concept and, when done well, makes music performance more fun and accessible to the general public.

To make the most out of the device's screen space, and as an aesthetic preference, we modeled only the parts of a violin that are essential for controlling a violin-like sound – the strings and a bowing region – and modified them to suit the iPad.

3.1.1 Strings

Instead of having four strings like a traditional violin, the Magic Fiddle has three, at intervals a fifth apart. Because strings are more difficult to reach as they lie farther away from the edge of the screen, we decided to use fewer strings. Also, we felt that the coordination required for four strings might overwhelm novice performers.

We tested two string layouts: vertical (long) along the right edge of the iPad, and horizontal (short) along the top edge. Strings in the center or across the diagonal of the iPad were considered, but rejected because they cannot be reached if the iPad is to be held like a violin. We decided on the vertical layout to offer more pixels per note for better pitch accuracy and fit more notes into a string. The final design has three strings fanning out from top to bottom along the right-edge of the iPad, with each string having a pitch range of one octave.

3.1.2 Bow Region

The intuitive gesture for playing a note is to touch the screen, with the note ending when the finger is lifted. However, it is not obvious how to associate a particular string with a particular touch. On a violin, the bowing angle determines which string is excited, such that players are free to place their fingers on strings that are not being bowed without affecting the sound. But on a flat screen device as an iPad, it is not possible to bow at a different angle, *per se*. We had to decide between two best alternatives: having three separate bowing regions each controlling a specific string, or having a single bow region that globally controls all strings.

The former option presents further complexity in the performance mechanics, as performers must touch the bow region corresponding to the string they wish to trigger. But it has an advantage of making open strings possible. For instance, if the player wishes to play D3 (the default fundamental of the lowest string), the player would simply touch the bow region corresponding to the lowest string with the right hand and not touch anywhere along the string with the left hand. Additionally, this option allows performers to put fingers down on strings that are not currently being bowed, in preparation for upcoming notes.

The latter option of having a single bow region offers simpler bowing mechanics, and a touch-on gesture on this general region would trigger all "active" strings. We generally do not

want all three strings to sound at all times, hence this approach assumes that a string is "active" only if it is touched at one or more points. Consequently, we can no longer play open-string nor put fingers down on a string that should not be sounding.

While the former option is closer to the mechanics of an actual violin and thus preserves much of the playing style on it, we chose the latter option for simplicity. We realized that fingering the notes correctly on the left hand presented enough technical challenge to the performers, especially considering that there is no tactile feedback in playing on an iPad. We wanted to provide users with a simple playing mechanics over a realistic representation of a violin.

3.1.3 Chin Rest

Initially, the plan was to make it mandatory for a user to place his or her chin on the corner of the screen (see Figure 2), but as it turned out, different chins seem to have varying result with the multi-touch (some would not activate, even with no facial hair). So eventually we moved away from this concept.

To reward users who are holding the iPad the "proper" (and possibly the more difficult) way, we reserved the bottom left corner of the iPad screen for the chin rest, animating bubbles when a touch gesture is detected there.

3.1.4 Musical Score

We used the remaining screen space to display a musical score for the "Songbook" mode (Figure 3). In a Songbook mode, the score is visually depicted as a series of line segments (each representing a music note) moving across the screen from left to right. This animation of incoming notes guide performers when and where to touch the string, and the color of the line segments guide which of the three strings should be fingered. Plucking interactions, vibrato, and glissandi are represented with different graphics. The idea of animating a score as a series of incoming objects is taken partially from the *Leaf Trombone* instrument, but the fiddle has an added complexity of having three "layers" of the score, one per string. [15]. Additionally, a particular track may present a piano accompaniment for the performer.

3.2 Audio Synthesis

We felt that the Magic Fiddle should support distinct sounds corresponding to bowing and plucking. We experimented with using STK Bowed instrument, as well as soundfonts and custom-synthesized sounds. Though the STK Bowed instrument allowed us to directly map bow position, bow pressure, and dynamics to the synthesis model, it was difficult to control these parameters using touch gestures given a relatively small bowing region that we had to work with. Also, naturally, certain combinations of physical parameters on the STK model do not produce pleasant sounds such that the performer may feel frustrated by the lack of responsiveness of the instrument synthesized using this method.

The various soundfonts we experimented with for plucking sounded quite rich and realistic, though soundfonts for bowing sounded artificial with their strictly-regular vibratos (or the lack thereof). So we continued experimenting with other synthesis techniques, in search of a more satisfying bowing sounds.

Eventually, we synthesized the bowed string using an implementation of commuted waveguide synthesis [11], which encapsulates the modeling of excitation, waveguide/resonator, body, and air responses. In the "classic" model, these components are meant to be processed in order, where one complexity is the digital filter required to model the body. Commuted synthesis takes advantage of the linearity and time-invariance of these components to combine the body and air components into a single impulse response that feed into the waveguide. This approach reduces the complexity of the system and is efficient to implement, requiring only an impulse responses and a feedback-delay component for each string. Ultimately, we also used this method to model the plucked interactions, as it resulted in a sound that is more consistent with the bowed interaction.

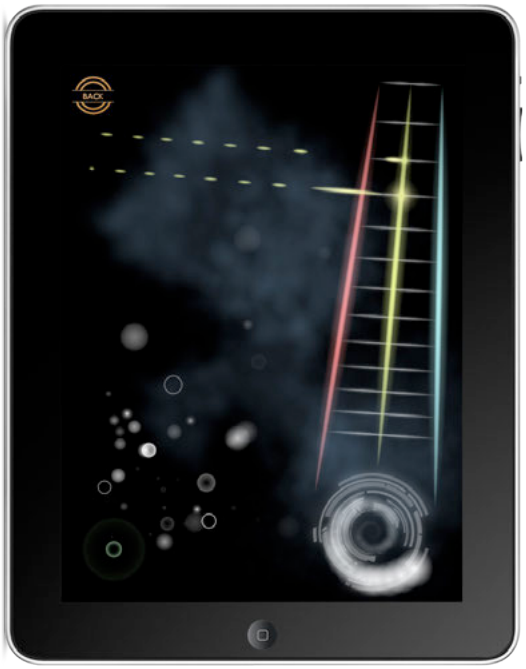


Figure 3. In Songbook mode, Magic Fiddle soft-prompts what and when to play using animated onscreen markers.

3.3 Customization for Playability

We wanted to offer flexibility in the design of our instrument, to accommodate varying levels and styles of playing. So we considered customizing various elements of the instrument, as described below.

3.3.1 Angle of string

The angle of string should be adjustable to accommodate different hand sizes. The angle would be tightened for smaller hands, and widened for larger hands.

3.3.2 Pitch snap

Performers should be able to choose the extent to which frequency "snaps" to the closest note. High pitch snap would help beginners with intonation, such that fingering on the string close to the note gets interpreted as fingering exactly on the note. On the other hand, turning off pitch snap would offer flexibility: advanced players can take advantage of this to perform with vibratos (as a pitch variation), or to play quarter tones, for instance.

3.3.3 Frets

Although violins do not have frets, the default rendering of the instrument would include fret-like horizontal bars drawn over the strings, showing the exact placement of each note on the string. Interestingly, we later found that professional classical violin players like to turn this feature off, relying on their ear to determine the exact positions that produce correct intonation.

3.3.4 Songbook Rate

The songbook rate should be changeable, to allow performers to practice songs at a slower tempo, for instance. The final implementation allows customizing the songbook playrate from four times as slow as the original tempo to twice as fast.

3.3.5 Solo key

The pitch of the lowest string to be used in the Solo mode can be set to any note between C2 and D4. When set to C2, the strings match the lower three strings on a cello, and when set to D4, the strings match the top three strings on a violin. Because the overall range of the instrument over the three strings is just over two octaves, allowing adjustments on the key makes it possible to cover a greater range of pitch, facilitating playing as an ensemble.

Figure 4. Customizing the fiddle.

3.4 Assisting Beginners

One of the original motivations for designing musical instruments on personal mobile devices is to make music more accessible to the general public; that is, people who do not necessarily consider themselves as "musicians". Thus, we wanted to ensure that the instrument is approachable for absolute beginners as well. Beyond the songbook mode, which allows users to choose songs of varying difficulty levels to practice and perform, we have included a "Storybook" mode for pedagogical purposes. In this mode, the personified fiddle, which the user names upon the initial launch of the application, becomes a teacher with a personality. This aspect of Magic Fiddle is discussed extensively in Section 4.

3.5 Establishing a "Flow"

If the Storybook is designed to welcome beginners to start playing the instrument, the Songbook mode offers a more competitive environment in which users polish their performance skills. Our design goal was to generate a flow experience, in which players are fully immersed in the performance with a strong motivation to improve. So we associated with the Songbook mode a game-like scoring and badge mechanism. At the end of each performance, the performer is shown accuracy (i.e. "38 of 42 notes hit") and the corresponding score (i.e. "Total Points: 38"). The individual performances are aggregated in a summary Profile, showing users statistics. A global leaderboard not only shows top-scoring users with highest aggregate points, but also includes the user herself to convey where she stands in relation to the top scorers.

4. A NEW MUSIC TEACHER

4.1 An Instrument with Personality

The first time a user launches Magic Fiddle, it plays a short tune and greets: "Hello. I am your fiddle." The text comes from nowhere; there is no avatar representing the fiddle. The iPad, which becomes a fiddle when the application is launched, introduces itself. A*nd it probably wants to be your friend.*

Throughout development, we referred to the fiddle's personality as "the voice of the fiddle." Much of what makes the fiddle likable is the way it "speaks" to the user, as if from one person to another. The voice engages users by implying a depth of character. Where motivation for self-improvement failed, the desire to connect with the fiddle would encourage users to spend more quality time with the instrument.

Since the voice was so important, Magic Fiddle exercises fine control over the way in which it is displayed. When the fiddle speaks, its words appear as white text occupying the left side of the screen. It uses complete sentences which start with capital letters and end in punctuation. Lines of text fade in at speeds which suggest calm, even speech. Though the timing and positioning are precisely specified by the writers, they vary throughout the app, because they were tuned by feel.

The fiddle speaks this way to appear intelligent, friendly, and warm. But that is only its predisposition. The fiddle can be a goofball. Sometimes it becomes preoccupied, boastful, lonely, or pleased. Its changing emotion is reflected in the writing, but also by breaking the rules described above. For example, the fiddle sometimes speaks very slowly to drive home an important point or express exasperation. When the fiddle gets excited, the text comes thick and fast. Sometimes the fiddle completes a thought, but thinks of a "zinger" and squeezes it in by scrolling everything else up by one line. In certain cases, the fiddle displays images and plays sounds to enhance its explanations. Even when its strings are silent, the fiddle has plenty of ways to be expressive.

4.2 Storybook: Integrated Teaching

The first time the user presses the "Storybook" button on the main menu, they are presented with the table of contents of "Your Very First Magic Fiddle Book". When they tap the first section header, "Holding Your Fiddle," the fiddle itself begins to teach them how to play.

This is an important mode. While other main menu options float in a general area, storybook's button is the keystone which holds the menu together, and it flashes. It is not an integrated tutorial which could be summed up in a few screens of text (which the user would surely skip). The storybook contains hours of content, covering skills as basic as holding the fiddle correctly (Figure 5), to advanced techniques like sordino.

4.2.1 iPad Fiddle Lessons

"Storybook" is actually a collection of books, each with a handful of chapters. There is one chapter per topic (for example, "Bowing" or "Upper Body Posture"). Each chapter follows a pattern: introduce a concept, practice some music which demonstrates it, then perform a piece which combines knowledge gained in this chapter with that of previous chapters. The chapter is split into sections so that the user can resume at any of these points of transition.

Once the user begins a story, however, the only level of hierarchy that matters is the book. The story flows from section to section, chapter to chapter, until the user reaches the end of the book and is sent back to the menu. The user falls into a rhythm, although they may not recognize what it is. They can back out and see the structure, but it doesn't matter in context. If the user is interrupted, the next time they tap the "Storybook" menu item, the book is opened to where they left off.

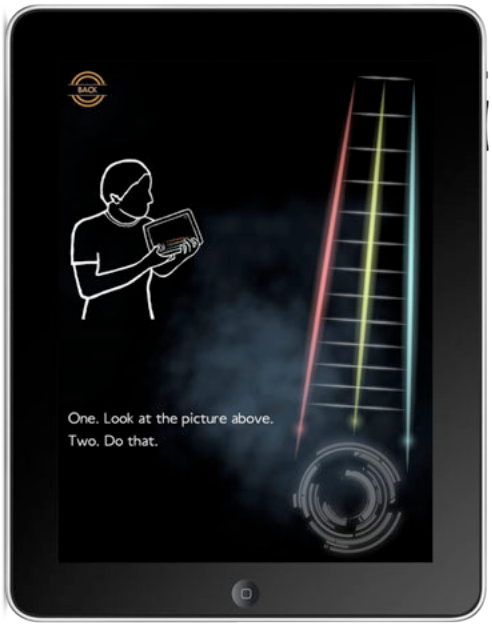


Figure 5. Magic Fiddle shows a user how it likes to be held.

4.2.2 Social Homework

Several of the chapters end with a mission for the user. These break up the normal lessons by asking the user to perform tasks which require interacting with people and places in the physical world. Although playing the fiddle by itself can be a rewarding experience, we wanted to encourage users to share the experience with others, face-to-face as a performative and social act.

After the completion of each mission, the user is asked questions about their experience. The responses indicate that gentle nudges from the fiddle were enough to inspire users to have a fantastic time with friends and family.

One mission asks the user to play "Mary Had a Little Lamb" while standing up to practice correct posture. Afterward, they are asked, "Could you sum up your experience in one word?" The top five responses were "Fun," "Awesome," "Cool," "Great," and "Nice." Here are some other notable responses:

"That was awesome. I am going to buy a real fiddle and practice what I learn from this app"

"爽" (translation from Chinese: "cool")

"A little bit harder than the actual violin"

"As a violinist, I can say its not even close to the real thing, but it was fun :)"

"The fiddle told me what to do. Awesome."

"It was so much fun. It was like playing my own violin!"

"I feel like a clown"

"Sounds like I'm making music!!!"

About 15 minutes of game time later, the user is asked to play "Twinkle, Twinkle, Little Star" in front of a live audience. The top five responses were again positive: "Fun," "Cool," "Great," "Awesome," and "Good." Some more responses are below:

"It was fun my audience (mom and dad) clapped"

"I was epic, the crowd cheered and lifted me up after I stage dived off my bed. Money and roses were thrown at me. It was pretty cool."

"Almost got a standing ovation from 2 dogs"

Unlike most of the storybook, which must be done in order, the user is free to skip missions. This gives users enough flexibility to comfortably perform the tasks required of them instead of faking their way through. This is more important for the later, more demanding missions. One asks the user to invite a friend (or potential lover) to a public play and play *L'amour est un oiseau rebelle*. Another asks them to busk outside a coffee shop while playing *Johnny Has Gone for a Soldier*.

5. IMPLEMENTATION

Magic Fiddle was developed using the Apple iOS Software Development Kit and additional third-party libraries including Fluidsynth[1] and the Mobile Music Toolkit (MoMu). The real-time graphical interface of the Magic Fiddle instrument was developed in OpenGL ES, and was optimized via additional visual cues to provide a natural feeling of fluidity and responsiveness.

Musical data (e.g., Songbook songs and Storybook snippets) is stored as MIDI files augmented with tags specifying the score for fiddle and piano accompaniment, as well as meta-data for pitch-to-string mappings and articulations. When rendering a Songbook or Storybook performance, a central scheduler synchronizes the graphics and audio and maintains a sliding window of upcoming notes. As noted in Section 3.2, the audio for the fiddle is an implementation of commuted waveguide synthesis. The piano accompaniment is rendered with soundfonts in a Fluidsynth. As in other Smule applications, Magic Fiddle presents a globe visualization that plays back performances of users playing the instrument around the world.

The applications tracks achievements (e.g., getting a perfect score on a song, or playing ten performances) and the user's position on a global leaderboard, ranked by total points earned.

The server-side components of Magic Fiddle provide storage and retrieval of user information, such as their storybook state, achievements; and maintains the global leaderboard. The latest version of all Songbook and Storybook information is stored and can be updated by the client application. Real-time telemetry data is collected from users as they interact with the application, providing opportunities for usability analysis.

6. REFLECTIONS

In the three months following its release, Magic Fiddle has been downloaded onto more than 100,000 devices. We are able to reflect on the user experience based on feedback from Storybook social homework responses, engagement data gathered via telemetry, reviews in the iPad App Store, and informal monitoring of Magic Fiddle-related posts on Twitter.

6.1 Customization for Playability

We provide an informal sample of tweets below to illustrate some different sentiments expressed about Magic Fiddle.

6.1.1 Enthusiasm

heatherlaforce: "Smule's magic fiddle is so much fun! Thankfully, it isn't too frustrating considering my musical training. It does strain the wrist however" (12/26/2010).

aprilynpodd: "I'm hooked on the iPad app: 'Magic Fiddle', Practiced for an hour today :)" (12/14/2010)

s**wimwims**: "Amazing experience! This holiday I've spent much time to play Vivaldi on my iPad:)" (11/23/2010)

cjkonecnik: "I just opened Magic Fiddle by Smule for the first time and when I exited, 2 hours had gone by!" (11/10/2010)

6.1.2 Nostalgia of Learning Violin

MildlyAmused: "It's like being in 4th grade again. My childhood, I am reliving it" (1/2/2011)

RedNinjaTurtl: "All those years of not practicing violin... now I'm gassed up about this $3 magic fiddle app. Ridiculous. god knows how much mom & dad spent" (12/7/2010)

[1] www.fluidsynth.org/

6.1.3 Social Engagements

timmmmyboy: "Ok the whole office is cracking up at my attempts playing the magic fiddle now" (11/18/2010)
Intenso: "this year's christmas eve I'll play Silent Night for my family on my #magic Fiddle" (11/26/2010)

6.1.4 Commentaries on Instruments

danhadi: "Love Magic Fiddle for iPad. It's a new breed of musical instrument" (11/24/2010)
heatherlaforce: "Smule Magic Fiddle is challenging, but so much fun! Perhaps I should get a real violin..." (12/24/2010)
e_2productions: "The magic ipad fiddle - an insult to REAL musicians, or a great tool to bring music alive to millions?" (11/18/2010)

6.2 Professional vs. Novice Performers

The members of the St. Lawrence String Quartet[2], an Ensemble in Residence at Stanford University, were among the first to try out Magic Fiddle. Their initial response noted the lack of tactile feedback on individual strings, and that the players missed the lateral curvature of the traditional instrument's neck. The ability to bow indefinitely was appreciated. Regardless of the many differences, these professional musicians picked up the instrument almost immediately, performing Pachelbel's *Canon* as an ensemble (Figure 6).

Figure 6. The St. Lawrence String Quartet on iPads.

In contrast, one of the authors had no experience playing a string instrument and at first began playing Magic Fiddle the "improper" way, by placing the iPad on a desk and using only the index finger to finger the notes. But after several hours of practice, she learned to hold the instrument properly, fingering with all four fingers on the left hand, and realized that this posture actually allowed her to improve her performance.

6.3 Concluding Remarks

The creation of Magic Fiddle was an experiment to craft a tangible artifact that is a *physical realization of a symbolic reality*. It combines physical metaphors of a violin with the virtual elements of a game and personal music teacher. While the preliminary response has been positive on the whole, we still have much to learn from the rich data it provides about how users relate to the experience. As the computer continues to evolve, and perhaps "disappear", we strive to find the right balance between leveraging its *physicality* and *virtuality*.

7. ACKNOWLEDGEMENTS

Thanks to the Smule team: Jeannie Yang, Tricia Heath, Ari Lazier, Mark Cerquiera, Michael Wang, Perry Cook, Nicholas J. Bryan, Jeff Smith, Scott Bonds, Jodi Ropert, Turner Kirk, Clara Valenstein, Rob Hamilton, Elon Berger, Erikk Lenfers, Nick Rudolfsky, Nick Kruge, Phaedon Sinis, Giancarlo Aquilanti, Renee Thomas, Sunil Pareenja, Jim Routh, Ribbit.

8. REFERENCES

[1] Bryan, N. Herrera, J., Oh, J., and Wang, G. 2010. "MoMu: A Mobile Music Toolkit." In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Sydney Australia.

[2] Cook, P. 2001. "Principles for Designing Computer Music Controllers." In *Proceeding of the International Conference on New Interfaces for Musical Expression*.

[3] Dahl, L. and Wang, G. 2010. "Sound Bounce: Physical Metaphors in Designing Mobile Music Performances." In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Sydney, Australia.

[4] Dourish, P. 2001. *Where the action is: the foundations of embodied interaction*: MIT Press.

[5] Essl, G. and Rohs, M. 2009. "Interactivity for Mobile Music Making", *Organised Sound*, 14(2): 197-207.

[6] Gaye, L., Holmquist, E., Behrendt, F., and Tanaka, A. 2006. "Mobile Music Technology: Report on an Emerging Community", In *Proceedings of the International Conference on New Instruments for Musical Expression*. Paris, France.

[7] Klemmer, S., Hartmann, B., and Takayama, L. 2006. "How Bodies Matter: Five Themes for Interaction Design." In *Proceedings Designing Interactive Systems*.

[8] Oh, J., Herrera, H., Bryan, N., Dahl, L., and Wang, G. 2010. "Evolving the Mobile Phone Orchestra." In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Sydney, Australia.

[9] Polanyi, M. 1967. *The Tacit Dimension*. London: routledge & Kegan Paul Ltd.

[10] Schiemer, G. and Havryliv, M. "Pocket Gamelan: Tuneable trajectories for flying sources in Mandala 3 and Mandala 4." In *NIME '06: Proceedings of the 2006 conference on New Interfaces for Musical Expression*. Paris, France.

[11] Smith, J. O., 2010. *Physical Audio Signal Processing*. W3K Publishing.

[12] Tanaka, A. 2004. "Mobile Music Making." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 154–156. Hamamatsu, Japan.

[13] Wang, G. 2009. "Designing Smule's iPhone Ocarina." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Pittsburgh, USA.

[14] Wang, G., Essl, G., and Pentinnen, H. 2008. "Do Mobile Phones Dream of Electric Orchestras?" In *Proceedings of the International Computer Music Conference*. Belfast, Ireland.

[15] Wang, G., Oh, J. Salazar S., and Hamilton, R. 2011. "World Stage: A Crowdsourcing Paradigm for Social / Mobile Music." *International Computer Music Conference*.

[16] Weiser, M. 1991. "The Computer for the 21st Century." *Scientific American Special Issue on Communications, Computers, and Network.*

[2] http://www.slsq.com/

MobileMuse: Integral Music Control Goes Mobile

R. Benjamin Knapp
Sonic Arts Research Centre
Queen's University Belfast
University Road
Belfast BT7 1NN
Northern Ireland, UK
b.knapp@qub.ac.uk

Brennon Bortz
Sonic Arts Research Centre
Queen's University Belfast
University Road
Belfast BT7 1NN
Northern Ireland, UK
bbortz01@qub.ac.uk

ABSTRACT

This paper describes a new interface for mobile music creation, the *MobileMuse*, that introduces the capability of using physiological indicators of emotion as a new mode of interaction. Combining both kinematic and physiological measurement in a mobile environment creates the possibility of integral music control—the use of both gesture and emotion to control sound creation—where it has never been possible before. This paper will review the concept of integral music control and describe the motivation for creating the MobileMuse, its design and future possibilities.

Keywords

Affective computing, physiological signal measurement, mobile music performance

1. INTRODUCTION

The relationship between emotion and music has become an obsession for researchers and popular culture over the past several years. With popular books such as *Musicophilia* [11] and *This is Your Brain on Music* [8] topping the best seller lists, it is evident that this topic has indeed a very broad appeal. The field covers topics ranging from musicology to psychology, and from social science to computer science. This paper will focus on one subset of this broad field—the concept of using direct physiological measurement of emotion to augment the existing kinematic and locative sensors in mobile phones to create a new form of group musical interaction. While research on the introduction of emotion as a component of human-computer interaction, so called *affective computing*, has been ongoing for many years (a good collection of articles can be found in [10]), the concept of integral music control, the capability to use both gesture and emotion in controlling musical instruments has been around a comparatively short time [3, 4, 9]. The concept of using emotion in a mobile music making environment takes integral music control outside of the standard performance environment and introduces new possibilities of musical interaction. This paper will review the concept of integral music control It will then describe the creation of a new interface, the *MobileMuse*, that enables integral music control as a new means of creating music in a mobile environment.

2. INTEGRAL MUSIC CONTROL

Integral Music Control (IMC) is defined in [3] as "a controller that:

1. Creates a direct interface between emotion and sound production unencumbered by a physical interface.
2. Enables the musician to move between this direct emotional control of sound synthesis and the physical interaction with a traditional acoustic instrument, and through all of the possible levels of interaction in between."

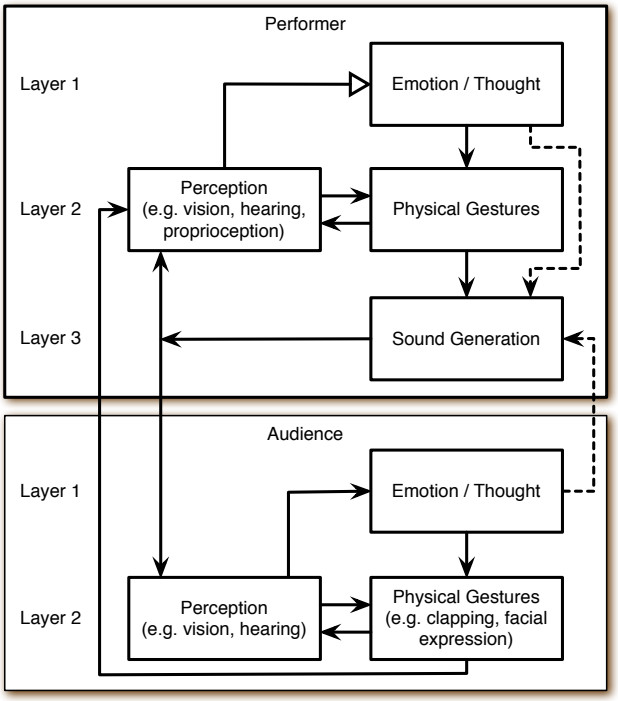


Figure 1: The three layers of performance feedback using IMC. Layer 1 represents the internal emotion and thoughts of the performer. Layer 2 is the physical interface layer. Layer 3 represents the consequence of the gesture—the creation of music. Performance feedback with the audience is also possible. The dashed line represents a new path of direct measurement of emotion. (From [3])

Figure 1 shows the standard technique of controlling sound generation: a thought creates a gesture which then controls a sound generator. Both the sounds and the proprioception of the physical interaction of creating the sound are then sensed by the performer creating a direct feedback loop.

The concept of integral music control opens up the possibility for the addition of direct measurement of emotion as another means of interaction.

2.1 Measurement of Emotion

To measure the emotional state and emotional changes of the performer, various physiological indictors of emotion can be used to achieve as accurate a measurement as possible while not interfering with performance. These signals include:

- *Galvanic skin response (GSR)* (Skin impedance)—Measured with electrodes on the finger tips
- *Electrocardiogram (ECG)* (Heart rate and heart rate variability)—Measured with electrodes built into a chest strap
- *Respiration (Amplitude and frequency)*—Measured with strain sensors built into an elastic chest strap
- *Electroencephalogram (EEG)*—Measured on the occipital (rear) portion of the head with electrodes attached to a head band
- *Facial electromyogram (EMG)*—Measured with sensors built into the same EEG head band

It should be made quite clear that these physiological indicators are not only measures of emotion. Indeed, as described in [10], there are many reasons other than emotional changes why these physiological signals might vary. However, the primary alternative reasons for variation such as changes in environment and changes in physical activity, do not apply in standard musical performance practice (even mobile) and consequently the reliability of these physiological signals as an indicator of emotional change is presumed to be high. As seen in Figure 1, the direct measurement of the audience's emotional state can also be used to interact with and even co-create with the performers. Members of the audience are, of course, not able to wear the large array of sensors worn by the performer. However, the audience can be connected to custom circuitry that measures GSR and ECG signals (see Figure 2). These two signals are chosen because of their capability to indicate changes in emotional state while still being easy to apply and relatively unobtrusive.

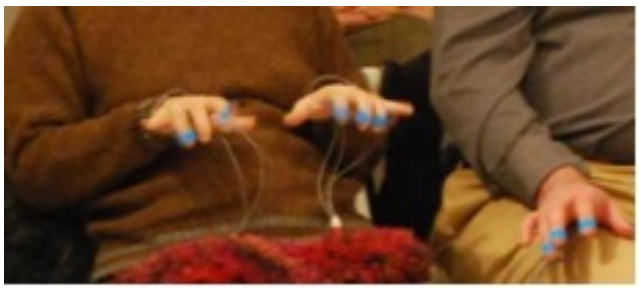


Figure 2: Sensors worn by audience members measure GSR and ECG. Heart rate and heart rate variability are then derived from the ECG.

2.2 IMC Systems for Live Performance

To implement true integral music control in live performance, a system is selected for the performer such as the *BioMuse*, which is composed of body worn sensors (both kinematic and physiological), that enable unencumbered movement during live performance. Figure 3 shows the data path for a complete performance configuration. Data from the BioMuse sensors worn on the performer, sensing both motion and physiological indicators of emotion, are transmitted through a Bluetooth link to a PC running the real-time signal processing software, EyesWeb. The processed data are then sent to Max/MSP via Open Sound Control (OSC).

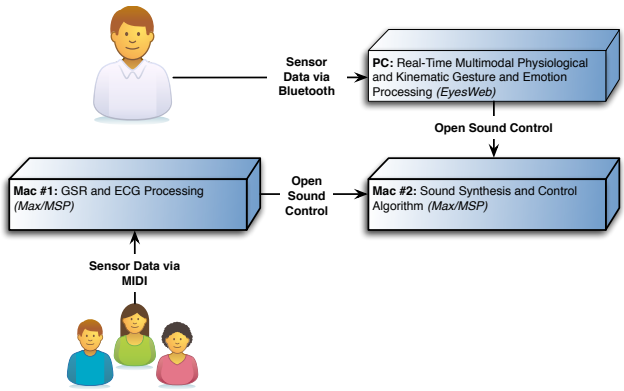


Figure 3: Data Flow for Sensor Acquisition and Processing.

The processed audience emotion signals are digitised and sent via MIDI to an Apple Mac running Max/MSP. All of the data from all of the audience members are sent via OSC to another Mac running Max/MSP. The reason for this separation of processing tasks into two computers is due to the DSP requirements of the primary sound synthesis Max/MSP patch. Separating the processing into two computers enables the creation of a dedicated GUI for the audience computer that allows real-time monitoring of the data and removal of any noisy data streams. Typically, in any performance there are always one or two audience members that remove their sensors or in some way manipulate the sensors so that their data are not useable.

3. MOBILE MUSIC CREATION AND THE MOTIVATION FOR A MOBILE IMC

Gaye, Holmquist, Behrendt and Tanaka define mobile music as "a new field concerned with musical interaction in mobile settings, using portable technology...[that] goes beyond today's portable music players to include mobile music making, sharing and mixing." [2] While not precisely mobile music as defined by Gaye et al., the concept of using mobile devices within an artistic context began, arguably, in 2001 in Golan Levin's work Dialtones [7], wherein audience members' mobile phones provided the sole sounding medium for the piece. As the ubiquity of mobile phones has grown worldwide (in June 2010, for instance, the number of mobile phones in Saudi Arabia grew to nearly double the country's actual population), so have the artistic community's pursuits of mobile music grown from niche diversions to full-scale areas of output and research. Performing ensembles have formed around the creation and concertising of new mobile music (e.g. [12]), and the sheer number of music generation and virtual instrument mobile applications that became available last year alone all point to the fact that mobile music and locative media as artistic (as well as entrepreneurial) endeavours are burgeoning phenomena.

The preponderance of currently available implementations of mobile music creation using mobile phones use only locative information, gestural information and/or other data gathered from the built-in sensors of the phones themselves.

Would there be advantages or new possibilities with the introduction of physiological assessment of emotional state in this new environment? Clearly, musical pieces that use audience emotional state measurement as a means of interaction, such as *The Reluctant Shaman* [5] or *Stem Cells* [6], would no longer need the custom hardware shown in Figure 2 to be installed before each performance. If there were a way to measure emotion using a mobile phone, then audience members could simply download the mobile phone application before coming to the concert (or even at the concert venue), connect the interface to the phone, and then participate in the performance.

While performers using IMC typically use more elaborate sensor configurations as described previously, a new type of less encumbered (both physically and spatially) mobile performance interface could also be imagined that would enable entirely new forms of interaction. Indeed it would be possible to implement the concept described in [4] and shown in Figure 4 to combine the physical gestures and emotional state of multiple performers before they are categorised and processed into control parameters. This would enable individual as well as composite measurements of gesture and emotional state. Both forms of networking can be combined to create a mesh of integrally networked IMCs. Thus, for example, a mobile performer's emotional state could be assessed by the IMC, combined with other performer(s) to create an overall combined emotional state.

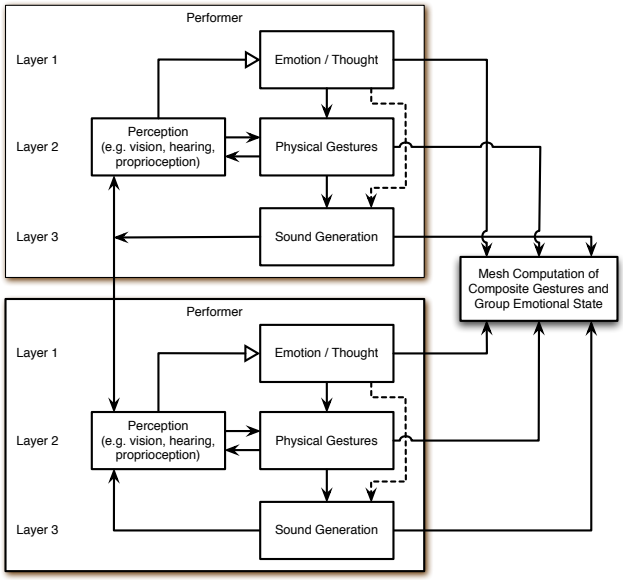


Figure 4: The networking of multiple IMC's using an integral control path as part of a distributed mobile environment. In this mesh, any performer's physical gestures and emotional state can be composited with any other's. (From [4])

4. THE MOBILEMUSE

In creating a device that could measure physiological indicators of emotion and interface to a mobile phone, two questions needed to be answered:

1. *What is the interface?* It is clearly desirable that any mobile IMC should be able to connect to any handset, not just one particular model. To fulfil this requirement, the answer to this first question is that the interface should be a standard TRRS audio jack. Rather than using the custom connectors that vary from mobile phone to mobile phone, this enables easy interfacing with most modern models. Indeed, this choice enables ubiquitous interfacing to most computing equipment (e.g. Macs and PCs) as well.

2. *What physiological signals would be used?* The answer to this question is subject to the same requirements as when measuring audience emotional state. It should be very easy to wear, but should also measure the key physiological indicators of emotion. Thus, as with the audience measurement hardware shown in Figure 2, GSR and heart rate were chosen as the physiological signals to be used for emotional state estimation.

4.1 Implementation

It was decided that the entire sensor system must fit on a single finger. In order to accomplish this, heart rate is measured using standard pulse oximetry techniques rather than a full ECG as was done for the audience of the IMC shown in Figure 2.

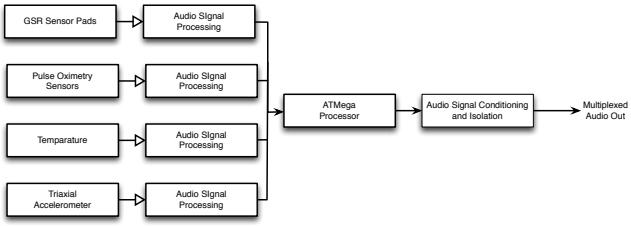


Figure 5: Block Diagram of the MobileMuse

As shown in Figure 5, four sensors are integrated into the MobileMuse. Upon further consideration, and because of the ease of design, a temperature sensor was also added to the interface. Skin temperature change (in relationship to the environment) has been shown to be indicative of long term mood [1] and it was thought that this might prove beneficial in assessment of emotional state. A triaxial accelerometer was also added to the circuit for gestural control. While this might seem redundant, it was thought that independent hand gesture might introduce interesting options not presently available with the accelerometers built into the phone. At minimum this enables two-handed gestural control.

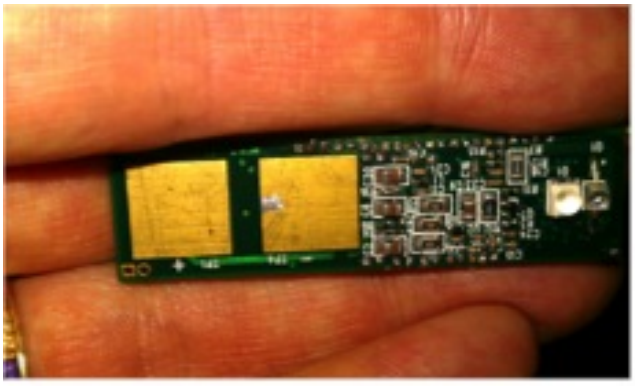


Figure 6: First implementation of the MobileMuse: The two large pads are for GSR measurement and the two LEDs on the far right are for pulse oximetry.

As shown in the block diagram in Figure 5, all of the sensor signals are amplified, processed, conditioned and then

connected to an ATMega processor. Choosing this processor means that the MobileMuse can be used as a custom Arduino board with all of the advantages this creates—most importantly, ubiquitous software availability. The ATMega processor is used to frequency-division multiplex the sensor signals in order to create one single audio data stream. The signal is then reconverted to an analog stream using the pulse-width modulation (PWM) output of the processor and subsequent signal conditioning. Finally, magnetic isolation is used to remove any shock risks and to eliminate line noise.

4.2 The iPhone App

The first requirement of the mobile phone application is to demodulate the frequency-division multiplexed signals coming from the MobileMuse. As shown in Figure 7, a simple application was created to implement this demodulation and display the waveforms and the composite estimate of emotional state derived from the physiological signals.

Figure 7: The MobileMuse application interface.

5. POSSIBLE APPLICATIONS

The MobileMuse enables a broad range of musical possibilities. Performers in a standard stage-based environment can now interact with audience members both locally and around the world. The MobileMuse audio interface enables stage performers to use low-cost wireless audio transmitters to connect to computers running sound creation software. Other BioMuse sensors can be connected to the MobileMuse in audio combiner mode so that any physiological or kinematic sensor can be connected via a standard audio interface. As with all mobile music creation, the concept of audience member and performer has lost all meaning. The emotional state of mobile listener/performers creates a feedback loop between listening and controlling the music being heard. This introduces one of the exciting possibilities of the MobileMuse which is to extend the realm of mobile music creation to the area of shared experience. As people across a mobile network listen to a music stream, watch a video, or even dance together, the emotional state of each listener or the composite emotional state of many listeners can be displayed visually or can introduce sonic changes representative of an individual or group as a whole. Of course, as discussed previously, there are always confounding factors in deriving changes in emotional state from physiology, e.g. one could be exercising or moving from indoors to outdoors. However, in the typical mobile performance scenario and with the performer's awareness of these issues, these factors can be largely mitigated.

6. CONCLUSIONS

The use of mobile phones as a new means of music creation and a new way of using location as a component of performance is fast becoming standard practice. This paper has introduced the concept and design of the MobileMuse, a new interface that enables direct measurement of physiological indicators of emotion to be used as a new means of interaction with a mobile phone. By bringing affective signal processing to a mobile platform, MobileMuse situates itself perfectly to marry three diverse arenas of artistic exploration that have, by and large, never been joined—mobile music, locative media, and affective creative practice.

7. REFERENCES

[1] T. Baumgartner, M. Esslen, and L. Jäncke. From emotion perception to emotion experience: Emotions evoked by pictures and classical music. *International Journal of Psychophysiology*, 60(1):34–43, 2006.

[2] L. Gaye, L. E. Holmquist, F. Behrendt, and A. Tanaka. Mobile music technology: Report on an emerging community. In *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, pages 22–25, Paris, France, 5–7 June 2006. NIME, IRCAM–Centre Pompidou.

[3] R. B. Knapp and P. R. Cook. The Integral Music Controller: Introducing a Direct Emotional Interface to Gestural Control of Sound Synthesis. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 4–9, Barcelona, Spain, 5–9 September 2005. ICMA, Phonos Foundation, Pompeu Fabra University and the Higher School of Music of Catalonia (ESMUC).

[4] R. B. Knapp and P. R. Cook. Creating a network of integral music controllers. In *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, pages 124–128, Paris, France, 5–7 June 2006. NIME, IRCAM–Centre Pompidou.

[5] R. B. Knapp, M. Dowling, and G. D. Ford. The Reluctant Shaman. Premiered at the 2008 International Computer Music Conference (ICMC), August 2008.

[6] R. B. Knapp and E. Lyon. Stem Cells–Reimagined. Premiered at the 2009 International Conference on Music and Emotion (ICME), August 2009.

[7] G. Levin. Dialtones (A Telesymphony), June 2001.

[8] D. J. Levitin. *This Is Your Brain on Music: The Science of a Human Obsession*. Dutton Adult, 2006.

[9] M. A. O. Pérez, R. B. Knapp, and M. Alcorn. Díamair: Composing for choir and integral music controller. In *Proceedings of the 2007 Conference on New Interfaces for Musical Expression*, pages 289–292, New York, New York, USA, 6–10 June 2007. NIME, Harvestworks and LEMUR.

[10] C. Peter and R. Beale, editors. *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*, volume 4868 of *Lecture Notes in Computer Science: Information Systems and Applications, including Internet/Web, and HCI*. Springer Berlin / Heidelberg, Heidelberg, Germany, 2008.

[11] O. Sacks. *Musicophilia: Tales of Music and the Brain*. Vintage Canada, 2008.

[12] G. Wang, G. Essl, and H. Penttinen. *Oxford Handbook of Mobile Music Studies (forthcoming)*, chapter The Mobile Phone Orchestra. Oxford University Press, London, England, UK, 2009.

Tangible Performance Management of Grid-based Laptop Orchestras

Stephen David Beck
Louisiana State University
Baton Rouge, Louisiana
sdbeck@lsu.edu

Chris Branton
Louisiana State University
Baton Rouge, Louisiana
branton@lsu.edu

Sharath Maddineni
Louisiana State University
Baton Rouge, Louisiana
smaddineni@cct.lsu.edu

ABSTRACT

Laptop Orchestras (LOs) have recently become a very popular mode of musical expression. They engage groups of performers to use ordinary laptop computers as instruments and sound sources in the performance of specially created music software. Perhaps the biggest challenge for LOs is the distribution, management and control of software across heterogeneous collections of networked computers. Software must be stored and distributed from a central repository, but launched on individual laptops immediately before performance. The GRENDL project leverages proven grid computing frameworks and approaches the Laptop Orchestra as a distributed computing platform for interactive computer music. This allows us to readily distribute software to each laptop in the orchestra depending on the laptop's internal configuration, its role in the composition, and the player assigned to that computer. Using the SAGA framework, GRENDL is able to distribute software and manage system and application environments for each composition. Our latest version includes tangible control of the GRENDL environment for a more natural and familiar user experience.

Keywords

laptop orchestra, tangible interaction, grid computing

1. INTRODUCTION

Laptop orchestras[8] (LOs) use an orchestral metaphor to provide an engaging and challenging environment to experiment with human-computer interaction, network and machine latency, and sound/signal processing. LO performers use ordinary laptop computers as instruments and sound sources for performing specially created compositions[7]. With the recent successes of the Princeton and Stanford laptop orchestras, LOs have now been established at many universities in the US, the UK, and as private ensembles around the world[11, 13].

LO composer-performers develop software to interpret human actions through computer interfaces that in turn control virtual instruments and processes that ultimately render music. Compositions can be improvised or scored, of determined or indeterminate length, with or without acoustic musicians. Laptops communicate across WiFi networks to synchronize time, distribute control messages, and manage other performance information.

Distribution, management and control of the necessary software across a heterogeneous collection of networked devices is a tremendous challenge for LOs. Each LO composition describes a potentially unique combination of core software, middleware and user-interface software that must be initialized, launched and performed. Configuration can range from the very simple (e.g., a single program on each machine, responding to keyboard events), to the very complex (Wii-motes, iPads, custom UI and laptops, driven by a networked time-sync). Laptops may all behave the same, or play specialized roles. The complexity of each piece and skill of each performer can affect the amount of time needed to prepare a piece for performance. And the "performance complexity" can scale exponentially with the number of laptops in the ensemble. Software may be stored in a central repository and distributed before a concert begins, but individual laptops must be configured and initialized immediately before the performance of each piece. Princeton's laptop orchestra identified software configuration as one of their most significant problems[9].

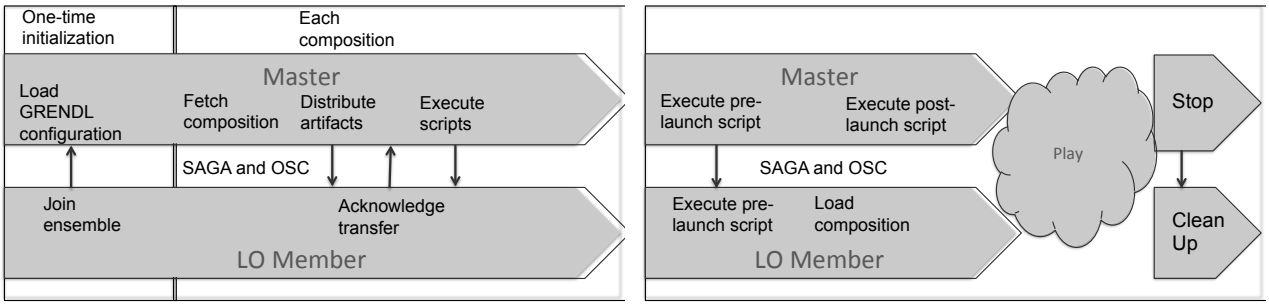
Our group has developed and field-tested the GRid ENabled Deployment for Laptop orchestras (GRENDL) system to help address the challenge of managing LO software distribution and configuration, while providing an experience for ensemble members that closely mimics that of a conventional orchestra. GRENDL is an integrated system that deploys, manages, and controls software and hardware technologies needed for the performance of music for laptop orchestras. For a given LO composition, GRENDL links digital artifacts (e.g., scores, software, electronic devices) with middleware applications (e.g., ChucK[12], Max[5], SuperCollider[4]) specific to the devices and operating systems available for performances.

2. LO PERFORMANCE WORKFLOW

A primary aim of GRENDL development is to support a workflow that is familiar to musicians. To maintain the orchestral metaphor, the system recognizes two distinct classes of ensemble computers. A single *master* machine, which is usually but not necessarily associated with the conductor, is responsible for loading and distributing the compositions, beginning and ending each piece, and managing the order of play. Any number of *performer* machines can join the ensemble, and play a specific role (i.e., instrument and part) in a specific piece.

The functionality provided by GRENDL can be seen as roughly analogous to the music librarian of a conventional orchestra, retrieving the parts for each musician and distributing them to the proper workstations. Some additional complexity is introduced in the LO case, since each laptop can (and likely will) serve as a different instrument for each

NIME'11, 30 May–1 June 2011 Oslo, Norway



(a) GRENDL transfer process

(b) GRENDL launch process

Figure 1: GRENDL supports a natural workflow between conductor (master computer) and orchestra members. (a) Before the performance, GRENDL loads the program and initializes the list of ensemble members. Digital artifacts are then transferred to each machine, ready for launch. (b) During the performance, GRENDL launches scripts to initialize each performer workstation, launch and configure the necessary middleware, and load the necessary data files. When the piece concludes, GRENDL launches scripts to restore each performer laptop to its default configuration.

composition. GRENDL distributes the appropriate music to LO musicians for a concert before the performance begins. GRENDL can also provide the conductor with the correct scores for each piece on the program, and give the conductor the ability to start and stop each piece in turn.

The entire program is transferred before the performance to minimize the delay between pieces, though the GRENDL architecture can support transfers at any time that a piece is not being played. The conductor can initiate a transfer command directly to the GRENDL engine using the command-line interface, or utilize the recently developed GRENDL Conductor application to manage the performance.

During the concert, the conductor instructs GRENDL to launch the software for each piece. GRENDL executes pre-launch scripts on the master computer to communicate roles, synchronize the ensemble, or perform other custom initializations, sends a "start" command to all computers in the orchestra, and runs post-launch scripts (if needed) on the master. Ensemble members use the GRENDL Performer interface to signal the conductor when their machine is properly configured and ready to play.

Once each laptop is configured and the proper software is launched, the piece is played. At the conclusion of each piece, the conductor triggers GRENDL's "quit" mode, which executes cleanup scripts on the machines.

The number of different scripts, configurations, hostnames, and other technical details that are needed to configure a LO can create a heavy cognitive burden on performers. "Cartouche" tangible user interface tokens[10] provide a simple and convenient way to encapsulate digital content and operations. GRENDL is designed to use cartouches to link client computers with specific performers and roles (e.g., percussion, voice). Cartouches may also be used to trigger software actions on the client computers, initiate messages to the GRENDL Conductor, or be linked to performance graphics for the musicians to use as a score.

3. GRENDL ARCHITECTURE

GRENDL has been created specifically to address the challenge of distributing and configuring software for LO performances. This is accomplished by viewing a LO as a computational grid [1]. One master machine, normally the conductor, assigns "jobs" to each of the remote performer nodes. In the case of GRENDL, the "transfer" jobs consist of using one of a variety of network protocols to distribute scripts, patches, program files, and other digital artifacts needed to play a composition. The "launch" jobs instruct the performer laptops to execute a series of scripts that configure the machine to play a specific composition. "Quit" jobs execute another script that stops any running software and cleans up any changes that were made to the environment.

The GRENDL software architecture includes components to manage connections to performer latops, retrieve and distribute the compositions, configure middleware and (when necessary) make system-level configuration changes, launch each composition, and restore the laptop meta-instruments to their default configurations after the piece has been performed. A conceptual overview is shown in Figure 2.

At the heart of the system is the GRENDL engine, a command line program written in C++ that is responsible for distributing and managing the jobs of each computer. The engine manages the transfer, launch, and quit jobs according to parameters specified in a set of configuration files associated with each piece in the program. The

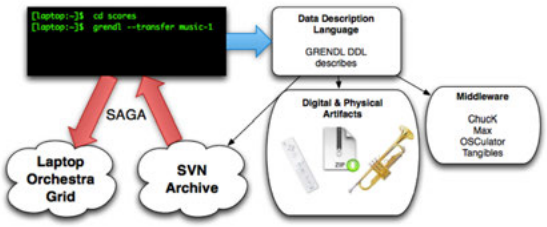


Figure 2: GRENDL includes components to retrieve compositions from online archives and distribute them to LO members. Other components load and configure middleware applications and compositions according to GRENDL data description language specifications.

current implementation of GRENDL uses the Simple API for Grid Applications (SAGA)[2] to manage the LO grid. SAGA is a grid computing framework that helps manage distributed applications in complex environments. SAGA connects application software written in C, C++ or Python with grid-based middleware services for distributed computing, providing a robust and platform neutral environment for complex computation. While SAGA was developed for high performance scientific applications using large grids of

hundreds or thousands of computers, it has adapted readily to the laptop orchestra environment. GRENDL leverages the SAGA framework for the distribution, initialization and launch control of LO software, treating the LO as a unique application of grid computing for live music performance.

An implementation of SAGA forms the file management and remote execution core of GRENDL, including support for configuring laptops, distributing compositions and supporting software, and launching the environment for each piece in the program. Since SAGA's synchronization is event-driven rather than time-based, it provides an ideal infrastructure for the asychronous and variable latency activities typical of performance preparation. These same features limit SAGA's utility during the performance, though SAGA still provides an important mechanism for file transfer and middleware configuration between compositions.

The most recent version of GRENDL uses Open Sound Control (OSC)[14], a protocol for communication among computers, sound synthesizers, and other multimedia devices, for more immediate and lightweight network communications. OSC is a simple, powerful protocol that provides everything needed for interactive control of sound and other media processing while remaining flexible and easy to implement. OSC libraries exist for most major programming languages, including C/C++, Java, Max[15] , and ChucK[12]. OSC interfaces have also been developed for a large number of interaction devices and visualization systems, as well as the majority of electronic instruments.

GRENDL uses OSC as the primary mechanism for intermachine communication during performances. This includes communication between the Conductor and Player applications, as well as distributing events generated by the tangible controls. OSC-based applications developed for the iPhone/iPad platform allow the conductor to set parameters for the ensemble and exchange information with performers. Max-based GUI's make use of OSC for communication between performers, and several ChucK pieces utilize OSC for synchronization and timing.

4. GRENDL CONDUCTOR

Conductor provides overall performance management for GRENDL. Conductor communicates workflow events to the members of the ensemble and manages the transitions from one piece to the next. It is the Conductor component that makes the calls to the GRENDL engine that will in turn deliver the jobs to the performers' laptops.

On startup, Conductor loads the program for the upcoming performance. Each item in the program represents a piece in the performance, including the location of the composition's digital artifacts. This location may be a folder accessible to the master computer, or the URL of an online repository.

Along with the program, Conductor loads a description of the ensemble, including account names and network addresses for all members. Before transferring the compositions to the laptops, Conductor listens for new members to join, and allows other members to be removed. Once the ensemble is complete, Conductor instructs the GRENDL engine to transfer each composition in the program to each orchestra member. After transferring all of the files, Conductor transitions to performance mode. By default, compositions are played in the order in which they are listed in the program, though this order can be changed through the user interface. For each piece, Conductor instructs the GRENDL engine to assign the appropriate "launch" job to each member of the ensemble. Once the ensemble is ready, Conductor enters "playing" mode. Until the signal is given to stop playing, Conductor will not send any other signals or process any remote requests. This minimizes the possibility that GRENDL will interfere with the performance of the piece.

The initial Conductor user interface was developed in Processing [6] to explore the capabilities needed to manage LO performance using GRENDL. It is expected that most or all of the interaction capability in Conductor will eventually be realized with a tangible user interface.

5. GRENDL PERFORMER

Complementing the GRENDL Conductor component is the Performer application, which is deployed on each orchestra member's laptop. Like Conductor, Performer was developed in Processing.

Performer provides an endpoint for communication between ensemble members and the Conductor. This allows ensemble members to register their laptops with the Conductor, thereby adding the machines to the laptop grid. Performer informs members of state changes in the performance, such as transfer and launch of specific pieces, and it allows orchestra members to signal the Conductor of specific events, such as when they are ready to play.

In addition to an OSC server, Performer monitors serial communications to detect events from the RFID cartouche readers. These events are translated into OSC messages and transmitted to the Conductor, except when a piece is being played. As with Conductor, Performer does not send events while a piece is being performed, and will only respond to the "quit" job sent from the Conductor.

6. TANGIBLE CONTROL OF GRENDL

The number of separate operations that must be performed to prepare to play a piece in a LO can be daunting. Configuring audio channels, loading middleware and data files, connecting and configuring external tools (e.g. Wiimote), and synchronizing multiple machines takes time, all while an audience watches. GRENDL helps address this problem, but at the cost of another layer of configuration, and another set of commands to be remembered.

Tangible user interfaces provide an effective answer to this new challenge. Specifically, cartouche tangibles[10] provide convenient tokens to represent concepts or actions. Cartouches can provide legible and actionable representations of compositions, performers, instruments, and programs that are usable by both human performers and electronic components of the LO. Cartouches provide a natural way for performers to interact with GRENDL, as well as a convenient and tangible representation of the elements of electronic music.

Cartouches have been tested with GRENDL in two contexts. First, a component has been developed that uses the Trackmate computer-vision based fiducial tracking system[3] to help configure the ensemble. When fully integrated, this will enable rapid and reliable reconfiguration of the orchestra for different pieces. The Trackmate system (Figure 3b) tracks the presence, position, and rotation of each cartouche, making a number of parameters available for future use.

LO members can be equipped with RFID tagged cartouches that are linked to the performer's identity within the group, machine information, or their specific role in a composition. In addition to concepts or entities, cartouches may represent actions or states, such as "ready to play."

7. CONCLUSIONS

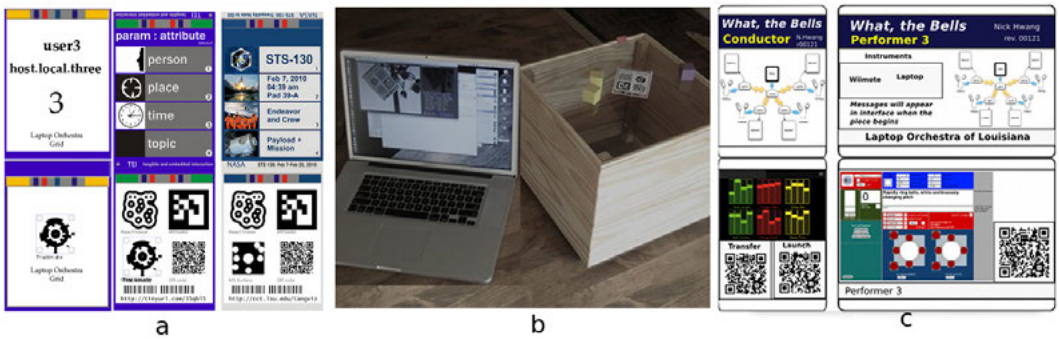


Figure 3: (a) Cartouche tangibles may represent a single concept, event, or role, or an entire set of values; (b) Trackmate is one of several computer vision systems that can provide a convenient way for LO members to interact with cartouches; (c) Cartouches are designed to be legible to humans as well as computers.

Our initial tests with the command-line version of GRENDL have demonstrated that it can be used successfully in a concert environment. It was initially piloted during a performance of the Laptop Orchestra of Louisiana (the LOLs) on April 14, 2010. Here, GRENDL was used to manage two LO compositions, and worked without incident. Over the following year, GRENDL was further tested on a regional tour by the LOLs, and managed an entire concert on April 4, 2011. We found that writing configuration settings was not nearly as straight forward as we wanted. Once tested and configured correctly, GRENDL worked flawlessly, and facilitated a very successful concert without any delays during the performance. We anticipate that the tangible version of GRENDL, which we will use in the coming concert season, will alleviate many of these issues.

These experiences have proven GRENDL's utility, confirming our belief that such a system addresses key impediments to the widespread adoption and long-term persistence of the laptop orchestra genre. That said, these tests have revealed additional parameters and actions that must be considered when building transfer, launch and quit scripts, especially in the realm of complex OS configuration.

Extending the GRENDL engine to integrate smoothly with tangibles, and in novel runtime environments, will require extensions to SAGA. The trans-disciplinary nature of GRENDL provides potential to shed new light on existing challenges in computational science. The LO setting presents a unique perspective from which to investigate topics such as time-sensitive and dynamic job scheduling, latency-bound interaction, and effective user interfaces for grid computing environments. Some of the first iteration interaction technologies have been developed for distributed computational science applications, and some of what is learned through GRENDL will likely be applicable in that area.

8. ADDITIONAL AUTHORS

Additional Authors: Brygg Ullmer (Louisiana State University, email: `ullmer@cct.lsu.edu`) and Shantenu Jha (Louisiana State University, email: `sjha@cct.lsu.edu`).

9. REFERENCES

[1] F. Berman, G. Fox, and A. Hey. *Grid Computing: making the global infrastructure a reality*. John Wiley & Sons Inc, 2003.

[2] T. Goodale, S. Jha, H. Kaiser, T. Kielmann, P. Kleijer, A. Merzky, J. Shalf, and C. Smith. A simple API for Grid applications (SAGA). In *Grid Forum Document GFD*, volume 90, 2007.

[3] A. Kumpf. *Trackmate: Large-scale accessibility of tangible user interfaces*. PhD thesis, Massachusetts Institute of Technology, 2009.

[4] J. McCartney. Rethinking the computer music language: Supercollider. *Computer Music Journal*, 26(4):61–68, 2002.

[5] M. Puckette. Max at seventeen. *Computer Music Journal*, 26(4):31–43, 2002.

[6] C. Reas and B. Fry. Processing: a learning environment for creating interactive Web graphics. In *ACM SIGGRAPH 2003 Web Graphics*, page 1. ACM, 2003.

[7] S. Smallwood, P. Cook, D. Trueman, and G. Wang. Composing for laptop orchestra. *Computer Music Journal*, 32(1):9–25, 2008.

[8] D. Trueman. Why a laptop orchestra? *Organised Sound*, 12(02):171–179, 2007.

[9] D. Trueman, P. Cook, S. Smallwood, and G. Wang. Plork: Princeton laptop orchestra, year 1. In *Proceedings of the 2006 International Computer Music Conference*, pages 443–450. Citeseer, 2006.

[10] B. Ullmer, Z. Dever, R. Sankaran, C. Toole Jr, C. Freeman, B. Cassady, C. Wiley, M. Diabi, A. Wallace Jr, M. DeLatin, et al. Cartouche: conventions for tangibles bridging diverse interactive systems. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 93–100. ACM, 2010.

[11] G. Wang, N. Bryan, J. Oh, and R. Hamilton. Stanford Laptop Orchestra (SLOrk). *Proceedings of the International Computer Music Conference*, pages 505–508, 2009.

[12] G. Wang, P. Cook, et al. ChucK: A concurrent, on-the-fly audio programming language. In *Proceedings of International Computer Music Conference*, pages 219–226. Citeseer, 2003.

[13] G. Wang, D. Trueman, S. Smallwood, and P. Cook. The laptop orchestra as classroom. *Computer Music Journal*, 32(1):26–37, 2008.

[14] M. Wright and A. Freed. Open sound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the 1997 International Computer Music Conference*, pages 101–104, 1997.

[15] M. Wright, A. Freed, and A. Momeni. Opensound control: State of the art 2003. In *Proceedings of the 2003 conference on New Interfaces for Musical Expression*, page 160. National University of Singapore, 2003.

Audio Arduino - an ALSA (Advanced Linux Sound Architecture) audio driver for FTDI-based Arduinos

as a demonstration of an open sound card system

Smilen Dimitrov
Aalborg University Copenhagen
Lautrupvang 15
DK-2750 Ballerup, Denmark
sd@{imi,create}.aau.dk

Stefania Serafin
Aalborg University Copenhagen
Lautrupvang 15
DK-2750 Ballerup, Denmark
sts@{imi,create}.aau.dk

ABSTRACT

A contemporary PC user, typically expects a sound card to be a piece of hardware, that: can be manipulated by 'audio' software (most typically exemplified by 'media players'); *and* allows interfacing of the PC to audio reproduction and/or recording equipment. As such, a 'sound card' can be considered to be a *system*, that encompasses design decisions on both hardware and software levels - that also demand a certain understanding of the architecture of the target PC operating system.

This project outlines how an Arduino Duemillanove board (containing a USB interface chip, manufactured by Future Technology Devices International Ltd [FTDI] company) can be demonstrated to behave as a full-duplex, mono, 8-bit 44.1 kHz soundcard, through an implementation of: a PC audio driver for **ALSA** (*Advanced Linux Sound Architecture*); a matching program for the Arduino's ATmega microcontroller - and nothing more than headphones (and a couple of capacitors). The main contribution of this paper is to bring a holistic aspect to the discussion on the topic of implementation of soundcards - also by referring to open-source driver, microcontroller code and test methods; and outline a complete implementation of an open - yet functional - soundcard system.

Keywords

Sound card, Arduino, audio, driver, ALSA, Linux

1. INTRODUCTION

A sound card, being a product originally conceived in industry, can be said to have had a development path, where user demands interacted with industry competition, in order to produce the next generation of soundcard devices. As such, the soundcard has evolved to a product, that most of today's consumer PC users have very specific demands from: they expect to control the soundcard using their favorite 'media player' or 'recorder' audio software from the PC; while the soundcard interfaces with audio equipment like speakers or amplifiers. For professional users, the character of 'audio software' and 'audio equipment' may encompass far more specialized and complex systems – however, the expectations of the users in respect to basic interaction with this part of the system is still the same: high-level, PC software control of the audio reproduced or captured on the hardware.

A development of a soundcard thus requires, to some extent, an interdisciplinary approach - requiring knowledge of both electronics and software engineering, along with operating system architecture. But, even with a more intimate understanding of this architecture, a potential designer of a new soundcard may still experience a 'chicken-and-egg' problem: understanding drivers requires understanding of their target hardware - *and* vice versa. As such, considering this product's origins in industry, it is no wonder that literature discussing implementations of complete 'soundcards' is rare - both hardware and software designs would have to be disclosed, for the discussion to be relevant.

An open soundcard. Businesses are, understandably, not likely to disclose hardware designs and driver code publicly; this may explain the difficulty in tracking down prior open devices. It is here that the Arduino [2] platform comes into play. Marketed and sold as an open-source product, it is essentially a board which represents a connection between a USB interface chip, and a microcontroller. As the schematics are available, an Arduino board can, in principle, be assembled by hand - however, a factory production has both a low, popular price; and brings in a level of expected performance, which allows for easier elimination of problems of electrical nature during development. Thus, on one hand, an Arduino board represents *known* hardware - one we could write an **ALSA** driver for; both in principle, and - as this project demonstrates - in reality. On the other hand, the Arduino is typically marketed as supporting communication speeds of up to 115200 bps (an impression also stated in [4]) - which result with data rates, insufficient to demonstrate streaming audio close to the contemporary CD-quality standard (stereo, 16-bit, 44.1 kHz). Yet, the major individual components: FTDI USB interface chip, and ATmega microcontroller - are both individually marketed to support up to 2 Mbps: a data rate that can certainly sustain a CD-quality signal. Thus, in spite of being *known* hardware, the Arduino may have 'officially unsupported' modes of operation, that would allow it to perform as a soundcard - modes that, however, still need to be quantified in the same sense, as if we were starting to design a board from scratch (*with this particular microcontroller, and USB interface chip*).

Application example. An open soundcard may bring actual benefits to electronic instrument designers, beyond the opportunity for technical study: consider a system where a vibrating surface (cymbal) is captured using a sensor and Arduino into **PD** software, where it is used to modulate a

NIME'11, 30 May–1 June 2011, Oslo, Norway.

digital audio signal in realtime. Usual approach would be to read the ARDUINO as a serial port at 115200 bps; this limits the analog bandwidth (≈ 5kHz) and forces the user to code a conversion to **PD**'s audio signal domain; with **AudioArduino** the sensor data could be received directly as a 44.1 kHz audio signal in **PD** - full audio analog bandwidth, no need for signal conversions.

2. PREVIOUS WORK

Previous attempts to discuss open soundcard implementations couldn't provide a basis for the development here: the Linux kernel contains many open soundcard drivers, but written for commercial (typically undisclosed) hardware. The now defunct german magazine Elrad may have had a series on implementation of a PCI card in 1997, but the remaining reference[1] doesn't contain any useful information. The ARDUINO has previously been used for audio: in [9] as a standalone player; [12] as a standalone DSP - but not specifically as a PC-interfaced soundcard. Thus, this project's basis is mostly in own previous work: [4] demonstrates legacy hardware controlled by PC software; and identifies data throughput control as the main problem in that naïve approach. Modern operating systems address this issue by providing a *driver architecture*; where, in programming a *driver*, the programmer gains a more fine-grained temporal control. In the context of the open **GNU/Linux** operating system(s), acquaintance with its current low-level audio library - **ALSA** - is thus necessary for implementation of soundcard drivers. This project has produced the tutorial driver **minivosc** [7] as an introductory overview of **ALSA** architecture - also used as a starting point of the work in this paper.

3. DEGREES OF FREEDOM

It would be interesting to qualify to what extent can **AudioArduino** - a system of ARDUINO Duemillanove, microcontroller code, and matching **ALSA** soundcard driver - be considered to be an 'open' 'soundcard system'. To begin with, hardware production necessarily involves mineral extraction and processing, manufacturing, and distribution - stages that require considerable economic infrastructure; and therefore, there will always be a 'hard' price attributed to it. On the other hand software, in essence, represents the instructions - information - for what we can *do* with this hardware. With the increasing affordability causing mass penetration of computing technology, fewer 'hard' investments need to be made to start with software development; and in principle, the pursuit of software development could thereafter involve only investment of the time of the developer. While developer time also carries inherent cost with it, there are circumstances where sharing the outcome - the source code - becomes preferable, for academic, business or altruistic reasons; especially since, with the expansion of the Internet, the physical cost of sharing information can be considered negligible.

Thus, it is in context of software that the term(s) 'free' or 'open' will be applied in this project (as in FLOSS[2]). To begin with, the driver is developed on **Ubuntu** - a FLOSS **GNU/Linux** operating system; with the main corresponding tool for development, **gcc**, being likewise open. The audio framework for **Linux**, **ALSA**, follows the same license - and the main high-level, user audio programs used, **Audacity** and **arecord**, are likewise open. The ARDUINO as a platform is known to be open, by making the schematic files available, as well as offering an integrated development environment (IDE) for **Linux**, which is also open [2]. The microcontrollers used in the platform are typically ATMEGA's, part of the ATMEL AVR family, which (given the tolerance of Atmel to open source, see Atmel Application note *AVR911*, also [14]) has long had an open toolchain for programming, **avr-gcc**.

At this point, let's note that ARDUINO in 2010 released the ARDUINO UNO board, which is taken to be the 'reference version' for the platform. The reason for this is that the USB interface chip used on the UNO is ATMEGA8U2, and the USB interface functionality is provided by the open-source **LUFA** (Lightweight USB Framework for AVR) firmware. In contrast, earlier versions of USB ARDUINOs, like the DUEMILLANOVE, feature a FTDI FT232RL USB interface chip. FTDI offers two drivers, VCP (Virtual COM Port, offering a standard serial port emulation) and D2XX (direct access) [18, 'Drivers']. Both of these are provided free of charge - however, source code is not available. Also, VCP may offer data transfer rates up to 300 kilobyte/second, while D2XX up to 1 Megabyte/second ([18, 'Products/ICs/FT245R']). Nonetheless, there exists a third-party open-source driver for FTDI in the **Linux** kernel, which corresponds to VCP, named **ftdi-sio** [11] - in fact, **ftdi-sio** forms the basis of the **AudioArduino** driver. With this, the following parts of the **AudioArduino** system can be considered open: *microcontroller code*, and tools to implement/debug it; *audio driver*, and tools to implement/debug it; *operating system*, hosting the development tools, the driver and high-level software; and *high-level audio software*, needed to demonstrate actual functionality – i.e., the bulk of the software domain. The driver was developed on **Ubuntu** 10.04 (Lucid), utilizing the 2.6.32 version of the **Linux** kernel; the code has been released as open source, and it can be found by referring to the home page [3].

4. CONCEPT OF AudioArduino

Given that the ATMEGA328 features both ADC, and DAC (in form of PWM), converters - using the ARDUINO as a soundcard hardware is a feasible idea, as long as one trusts that the data transfer between the PC and the ATMEGA328 can occur without errors at audio rates. Developing a USB driver for such data transfer would, essentially, require a good working knowledge of the USB bus and its specifications. However, that is a daunting task for any developer - the USB 2.0 Specification [19] alone is 650 pages long; with actual implementation, in a form of a driver for a given OS, requiring additional effort. Therefore, the starting point of this project is to abstract the USB transport to the greatest extent possible, and avoid dealing with particular details of the USB protocol. This is possible because of the particular architecture of the ARDUINO board, rendered on Fig. 1.

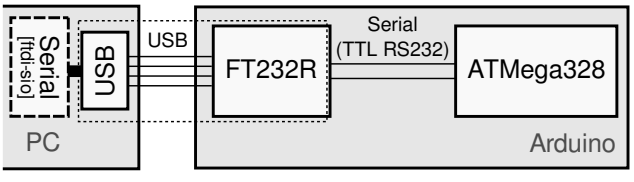


Figure 1: Simplified context of an ARDUINO, connected to a PC.

As Fig. 1 shows, the **ftdi-sio** driver makes the FT232 device appear as a 'serial port' in the PC OS, that the user can write arbitrary data to. The driver will format this data as necessary for USB transport, and send it on wire; the FT232 will then accept this data and convert it

[1] http://www.xs4all.nl/~fjkraan/digaud/elrad/pcirec.html
[2] free/libre/open source software

to TTL-level (0-5V) RS-232 signal (and the same happens for the reverse direction, when reading). Given that RS-232 is conceptually much easier to understand (e.g., [5]); we can 'black box' (abstract) the *unknown* (USB) part in the data transfer - and focus on the *known* (RS-232) part.

In order to specify what sampling rates, in terms of digital audio, would this hardware support - the most important factor to consider is the data transfer rate, that can be achieved between the ATmega328 and the FT232 over the serial link. As far as this serial link goes, the ATmega328 states maximum rate of 2.5 Mbps [16, pg.199]; while the FT232 states up to 3 Mbaud [17, pg.16]. As the **ftdi-sio** driver supports 2 Mbps[3] by default, this is the 'theoretical' speed that should be possible to achieve all the way through to the ATmega328. A speed of 2 Mbaud translates to 200000 Bps[3], which would be enough to carry 200000/44100 = 4.5 mono/8-bit/44.1 kHz channels; or two mono/16-bit/44.1 kHz channels; or one CD quality stereo/16-bit/44.1 kHz channel. However, one still needs to determine what *actual* data transfer rates can be achieved, and under which conditions (such as different software). Beyond this, it is the response times of the ATmega328 (including DAC and ADC elements), that would limit the use as full-duplex device. The final issue is the analog I/O interface, discussed further in this paper.

Building and running. Both the source code, and instructions for building and running, can be found in [3] (and they are similar to those given in [7]). The source code consists of a modified version of [11], **ftdi_sio-audard.c**; the **ALSA**-specific part in **snd_ftdi_audard.h**; associated headers and a **Makefile**; and microcontroller code, **duplexAudard_an8m.pde**. The `.pde` code can be built and uploaded to the Arduino using the **Arduino IDE**.

With this in place, high-level audio software (like **Audacity**) will be able to address the Arduino, and play back and capture audio data through it. Arduino's analog input 0 (`AIN0`) is treated as a soundcard input; sensors (like potentiometers) connected to this input can have their signal captured at 44.1 kHz in audio software. Arduino's digital pin 6 (`D6`) is soundcard output; on which, when audio software plays back audio data, (analog) PWM output is generated (audible).

5. QUANTIFYING THROUGHPUT RATE - DUPLEX LOOPBACK

As mentioned, one of the biggest issues in estimating if the Arduino board can behave as a soundcard, is in measuring the actual data transfer rate that can be achieved. The initial question is what tools can be used for that: the **ftdi-sio** driver will make a connected Arduino appear as a special file in the **Linux** system (`/dev/ttyUSB0`), representing a serial port. The serial port settings, such as speed, can be changed by using the **stty** program. Thereafter writing character data to the Arduino can be performed by writing to the associated file, say, by using `echo 'some text' > /dev/ttyUSB0` - and reading by, say, `cat /dev/ttyUSB0`.

However, finding the actual data rate in either direction is not the only thing which is interesting; another interesting point is to what extent can the Arduino board be considered a *full-duplex* device; i.e., whether the device can both receive and send data *simultaneously* (which, in terms of soundcards, is a standard expected behaviour). To assess both points, we suggest the ATmega328 is programmed as a 'digital loopback': to listen for incoming serial data; and send back the received byte through serial, as soon as it has been received. Then for the PC side, we propose a simple threaded program, **writeread.c** [15]: it accepts an input file; initiates write and read operations on a serial port in separate threads, so they can run concurrently; writes the input file, and saves the received data in another; and times these operations, so that the throughput rate can be determined.

What this experiment shows, is that the usual `C` commands for reading and writing from a serial port (and by extension, user programs like `cat` or `echo`) do not carry the concept of a data rate - they simply try to transfer data as fast as possible; and even for 2 Mbps communication, these commands push data faster than the USB chip can handle, which results with kernel warnings. Therefore, it is up to the program author to implement some sort of buffering, that would provide an effective throughput rate. Yet even with this in place, limiting rate to 2 Mbps within **writeread.c** would *still* cause throttling warnings; but, limiting it to slightly *below* 2 Mbps allows for a error-less demonstration. The reason for this is likely in the asynchronous nature of the serial RS232 protocol: in not sharing a single clock; the PC, the FT232 and the ATmega328 each have a slightly different concept of what the basic time unit (clock tick) duration would be - and thus a different concept of what '2 Mbps' is. By lowering the data rate from **writeread.c**, we likely account for these differences, which allows for error-free transmission; and from the PC, we can typically measure around 98% of 2 Mbps achieved for error-free duplex transmission.

Moreover, during this digital loopback experiment, the signals of the `TX` and `RX` connections (between the FT232 and the ATmega328) were measured with an Agilent 54621A[4] oscilloscope; captured with the open-source **agiload** for **Linux**; and analysed using a script produced by this project, written in **python** (utilizing **matplotlib**) that features a serial decoder, called **mwfview-ser.py** [3]. These measurements show that the time for the ATmega328 to receive a byte and send it back - the minimal 'quantum' of action, relevant for a 'digital duplex' - is around 6.940 µs (Fig. 2), which is approx. 31% of the 22.6 µs analog sample period (for 44.1 kHz rate); which specifies the latency bottleneck expected from the Arduino in 'digital loopback' mode.

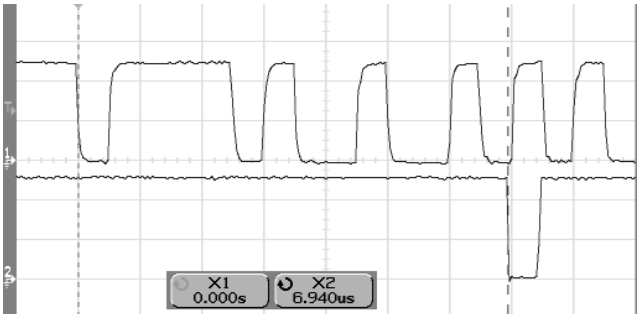


Figure 2: Oscilloscope capture of RX (top) and TX (bottom) serial lines at the ATMega328, indicating latency between received and sent byte.

Note that, the ATmega328's UART produces a signal

[3]Note that in 8-N-1 RS232 transfer, there are 8 data bits, 1 start and 1 stop bit; so 8-bit data is carried by 10-bit packet. Usually, 'baud' means 'signal transitions per second' and refers to all 10 bits, while 'bps' as 'bits per second' should refer to the 8 data bits only; but they can be often used interchangeably - 'Bps' as 'bytes per second' refers strictly to data payload (see also [15]).

[4]The Agilent 54621A claims 60 MHz bandwidth, which is sufficient for capture of a 2 Mbps digital signal

with considerably more jitter than the FT232[5]; and there can be gaps in the otherwise sustained rate of serial transmission between the two - but none of this seems to harm error-free transmission at 2 Mbps. Finally, **writeread.c** works both with the 'vanilla' **ftdi-sio** driver, and the **AudioArduino** driver. Also, the same ARDUINO code used to demonstrate digital loobpack with **writeread.c**, can be used with the **AudioArduino** driver - allowing for demonstration of a *digital audio loopback*: one can load a file in **Audacity**; play it back through the **AudioArduino** card; and by recording at the same time from the same card, one should capture the very same audio being played back (latency notwithstanding).

6. MICROCONTROLLER CODE

There are two distinct versions of microcontroller code for the ATMEGA328 used in this project, both in a form of a `C` language `.pde` file (the default format compilable in the ARDUINO IDE). The first is the mentioned 'digital duplex' code, which simply sends back any byte received through serial, posted in [15]. The main issues here are: the setup of the ATMEGA328's UART to support 2 Mbps (which is not supported in the default ARDUINO API); removing all overhead due to API function calls, by using the function source code directly; and disabling all irrelevant interrupts - before the ARDUINO can start showing 98% of 2 Mbps with **writeread.c**. Beyond this, the code can be implemented either as a single loop, or with interrupts on incoming serial data; with no significant difference in respect to performance. This is the same microcontroller code used as basis for development of the **AudioArduino** driver.

Once the **AudioArduino** driver was confirmed to be working with the 'digital duplex' code - a new, second 'analog I/O' version was written, which also employs the ADC and PWM (as DAC) facilities of the ATMEGA328. This version, as it is supposed to support audio playback and recording, requires deeper involvement with the ATMEGA328 datasheet [16]. In essence, the problem is that **ALSA** will send (mono) data at rate of 44100 Bps, which will appear as chunks of bytes on the 200000 Bps serial ARDUINO line; these bytes need to be stored as soon as possible by the ATMEGA328 in memory (buffer). On the other hand, at a rate of 44100 Hz, the ATMEGA328 should read one byte from the buffer and write it to PWM (the DAC) - and at the same time, read a byte from the ADC, and send it via serial. As we would expect an 8-bit interface (where each byte represents an analog sample) at the driver side, no further digital sample processing needs to be done in either direction. This is solved by code that employs an interrupt on incoming data, where the data is stored in a circular buffer - and a (16-bit) timer interrupt to handle the analog I/O at the 44100 Hz analog rate [3]. Note that this 'analog I/O' version seems to only perform well when implemented with incoming data handled on interrupt; trying to do the same handling in a single loop reveals problems with determining *when* an incoming byte is ready to be read from ATMEGA's UART [3].

7. DRIVER ARCHITECTURE

The **AudioArduino** driver is not only based on **ftdi-sio** - **ftdi_sio-audard.c** is a renamed version of [11], with several changes: first, it includes **snd_ftdi_audard.h**, which here is not used in the standard sense of a `C` header, but simply as a container for **ALSA** relevant code (which would, otherwise, have to be written into the already complex [11]). Other changes include calling **ALSA** relevant functions from the default **ftdi-sio** functions: `audard_probe` from `ftdi_sio_probe`; `audard_probe_fpriv` from `ftdi_sio_port_probe`; `audard_remove` from `ftdi_sio_port_remove`; and `audard _xfer_buf` from `ftdi_process_packet` - which connects the soundcard **ALSA** interface to USB events.

Otherwise, the main **ALSA** functionality is contained in **snd_ftdi_audard.h**, whose development is based on **minivosc.c** [7]. Thus, it contains the same type of **ALSA** related structures, but the structure map (shown on Fig. 3) is slightly more complex than in [7]: the main 'device struct', `audard_device`, contains an array holding references to both the playback and the capture substream; the substreams are encapsulated in `snd_audard_pcm` structures, that hold individual buffer position counters. There are separate `snd_pcm_hardware` and `snd_pcm_ops` variables - yet a single `snd_card_audard_pcm_timer_function` - to handle the playback and capture substreams.

In essence, the **AudioArduino** driver leaves, for the most part, the functionality of **ftdi-sio** as is; with several additions. When `ftdi_sio_probe` runs (i.e., when the ARDUINO is connected to PC via USB), the **ALSA** interface is additionally setup, enumerating the ARDUINO as a soundcard. With this in place, on one hand, the driver keeps the serial interface (such as the creation of the `/dev/ttyUSB0`) file. On the other hand, the driver will also react on 'start' or 'stop' commands from high-level audio software as usual: e.g., on 'start' `_trigger` will run, which will start the timer, and thus the periodic calls to `_timer_function`. The `_timer_function`, then, needs to handle the playback direction by copying the respective part of its `dma_area` to USB - which it does by calling `ftdi_write`. For the capture direction, incoming USB data triggers `ftdi_process_packet`, which additionally calls `audard_xfer_buf`; here USB data is copied to a dynamically sized 'intermediate' buffer, `audard_device->IMRX` – and `_timer_function` will thereafter copy the data from the intermediate buffer to the capture substream's `dma_area`, the next time it runs.

The **AudioArduino** driver additionally exposes CD quality, stereo/16-bit/44.1kHz capability - to allow for direct playback interface with **Audacity** (and most media player software). However, since the microcontroller code expects a sequence of 8-bit values, we must convert the stereo 16-bit stream to a mono 8-bit one - this opens a whole new set of problems related to wrapping, which is illustrated on Fig. 4. By declaring the driver capable of 16-bit stereo, we have not changed the number of substreams (which would correspond to connectors on the soundcard); however, Fig. 4 shows that we would have changed the data format carried in the substream's `dma_area` - the stream is now interleaved: consecutive bytes carry a pattern of left channel's 2 bytes, followed by right channel's 2 bytes. Thus an **ALSA** *frame* (size of analog sample in all channels) is now 4 bytes; and the problem becomes how to represent this **ALSA** frame with a single byte. The approach in the **AudioArduino** driver is to simply extract the most significant byte of the left channel, according to the formula (`C` code):

```
(char) (left16bitsample >> 8 & 0b11111111) ^ 0b10000000
```

However, as Fig. 4 shows, a bigger problem is that the wrapping boundaries (at the size of the chunk handled at each `_timer_function`, and at the size of `dma_area`) can now occur in the *middle* of a frame (and correspondingly, middle of an 8-bit sample) - which is a situation that doesn't occur for 8-bit streams (where each single byte corresponds to one analog sample). To address this, the **AudioArduino** driver employs yet another intermediate buffer (`audard_device->tempbuf8b`). With this in place, the driver will automat-

[5] A crude measurement of jitter spans around 0.26 µs, which is about 52% of the 0.5 µs period for a bit transition at 2 Mbps, see [15]

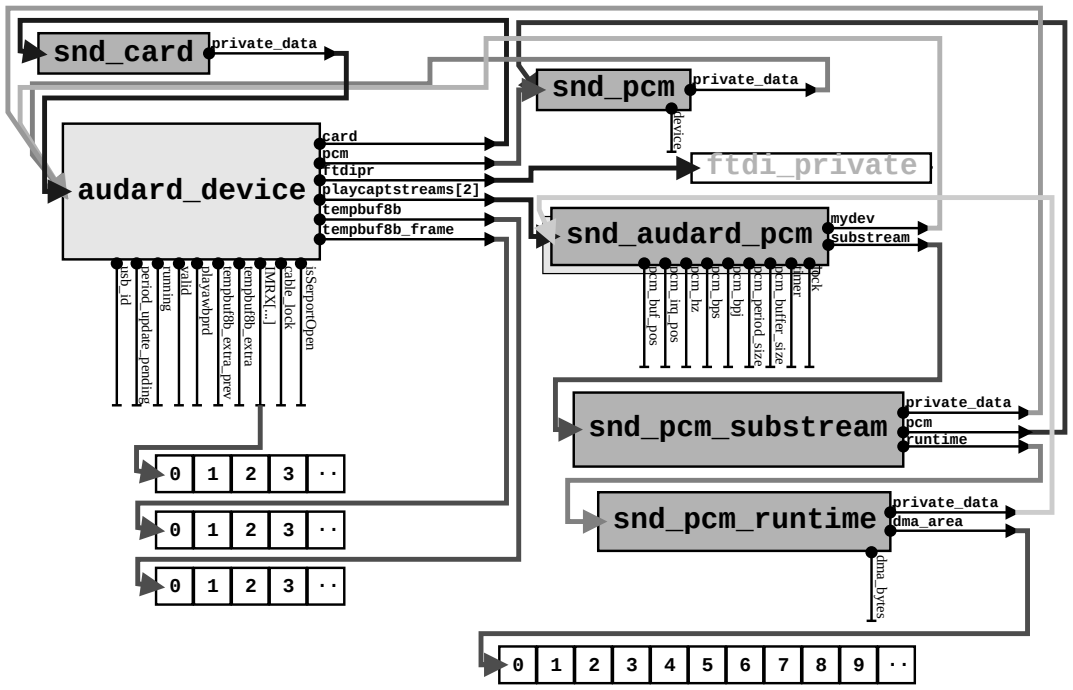


Figure 3: Partial 'structure relationship map' of the AudioArduino driver.

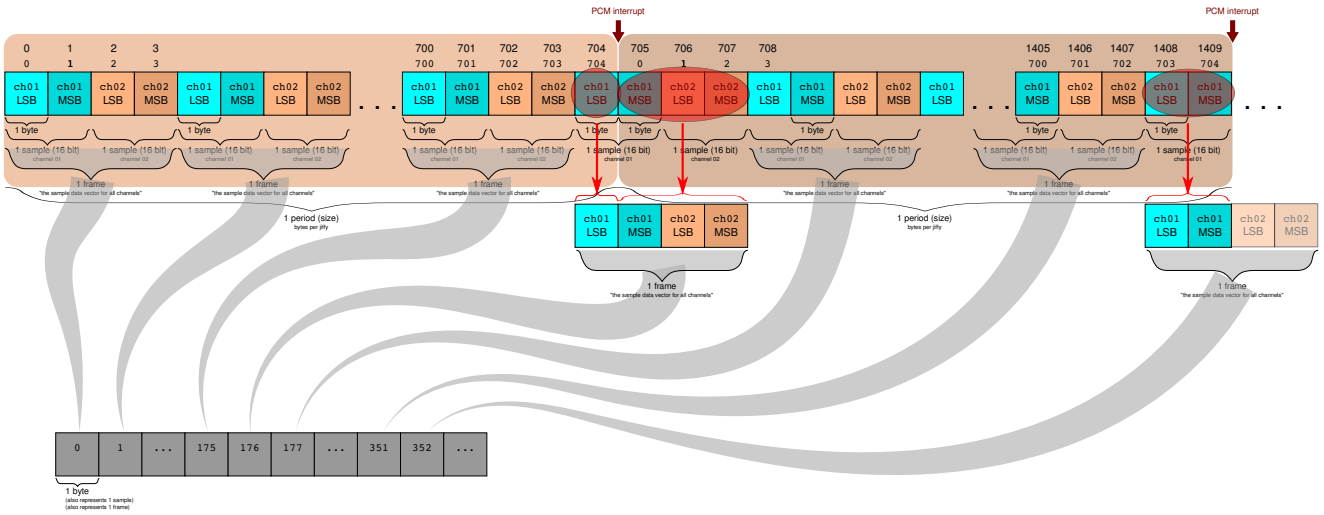


Figure 4: Visualisation of driver's playback buffer boundaries, and CD to mono/8-bit conversion.

ically convert a 16-bit stereo stream from **Audacity** to an 8-bit one, preserving the 44100 Bps rate, before it sends it to USB - and thus, an audio 'digital loopback' can be demonstrated on this driver directly from **Audacity**.

Finally, note that 'DMA' in '`dma_area`' stands for 'Direct Memory Access', which "*allows devices, with the help of the Northbridge, to store and receive data in RAM directly without the intervention of the CPU (and its inherent performance cost)* [8]". Interestingly, in this case: while the transfer of incoming USB data to PC memory (as part of **ftdi-sio**); as well as the transfer of data from `dma_area` to user memory of high-level audio software (as part of the **ALSA** 'middle layer'); likely involves DMA – the transfer of memory that is performed as part of **AudioArduino**'s `_timer_function` definitely *doesn't*; as we use the `memcpy` command to transfer data (which does involve the CPU).

8. ANALOG I/O

The **ALSA** driver can be developed in its entirety with the 'digital duplex' ARDUINO code; if thereafter the 'analog I/O' microcontroller code is 'burned' on the ARDUINO - the driver will, effectively, utilize analog input pin 0 as analog input connector, and digital pin 6 as analog output connector. However, both the analog input range, and the output PWM signal, span the voltage range from 0 to 5V - while a typical off-the shelf soundcard typically contains 'line' input and output connectors, as well as 'mic in' and 'speaker out' connectors, which follow a different analog standard. These topics are discussed in more detail in an associated paper, [6].

The use of analog pins on the ARDUINO to read sensors is standard practice, and plenty of examples can be found on the web [2]; thus an arbitrary sensor signal can be captured through high-level audio software at 8-bit, 44.1kHz quality (in the same spirit of [4]). Note that the analog input voltage range, 0-5V, will be represented with the span of 8-bit values from 0 to 255 - which within **Audacity** may be treated as floating point values -1 and 1, respectively.

The use of PWM to deliver an analog audio signal is based

on the premise that the highest PWM frequency obtainable from the ARDUINO, 62500 Hz [6], will be sufficient to reproduce a 44100 Hz digital (22.05 kHz analog) audio signal. To a novice, used to analog voltage waveforms, this can be problematic to assess - as the binary nature of PWM makes it seem inherently 'distorted' in the time domain. However, industry insiders are well aware of the practice of using PWM for audio, e.g., in the mobile or automotive industry [10], and often to drive speakers directly [1]. This project demonstrates that as well: upon playback of audio from high-level software, one can simply connect the output pin 6 to a channel on headphone jack, and connect the ground of the headphone jack to ARDUINO's ground - and audible sound would be perceived from the headphones' speaker (but use of a capacitor will result with a louder, clearer sound [3]). Note that there are inherent jitter problems in reproducing HF tones with this technique, while mid-range music can be reproduced with acceptable quality [3, 6].

9. CONCLUSIONS

As this paper outlines, development of a soundcard can be a complex and involved issue. The particular approach used here, avoids many electronic engineering issues by choosing the ARDUINO DUEMILLANOVE as soundcard hardware; and avoids deeper involvement with the USB protocol by the specific use of the **ftdi-sio** driver as a basis. In doing that, the overview of the **ALSA** architecture, started in [7], is finalized - as **ALSA** is discussed in its full intended scope: in relation to a given soundcard hardware, and given interface bus. This allows for focus on issues in soundcard implementation that are close to 'first principles', and as such could serve in educational context, as a basic introduction to newcomers to the field - which is the main contribution of this paper and source code.

Beyond (hopefully) furthering the discussion on DIY implementations of PC interfaced digital audio hardware, this project may have a practical impact as well - as there are research projects in the computer audio community and related fields (such as haptics [13]), which use the ARDUINO to capture sensor data; and as such, could benefit from the audio-rate capture quality, and the possibility to leverage the real-time performance of applicable high-level audio software, such as **PureData**.

10. FUTURE WORK

The current **AudioArduino** code could, in principle, easily be modified to demonstrate stereo 8-bit performance, or even 16-bit mono (say, by using separate PWM for LSB and MSB, and mixing them in the analog domain). A more involved work would be to port the concept to the reference ARDUINO UNO - as that will require work on the **LUFA** firmware, which doesn't currently support 2 Mbps[15]; on the other hand, the **LUFA** could allow the ARDUINO to be recognized as a 'USB audio' class device, instead of a 'USB serial' one. Finally, as in [7], it would be interesting to see to what degree could **AudioArduino** be ported to the major proprietary PC operating systems.

11. ACKNOWLEDGMENTS

The authors would like to thank the Medialogy department at Aalborg University in Copenhagen, for the support of this work as a part of a currently ongoing PhD project.

12. REFERENCES

[1] F. T. Agerkvist and L. M. Fenger. Subjective test of class d amplifiers without output filter. In *117th Audio Engineering Society Convention*, 2004.

[2] arduino.cc. Arduino homepage. http://arduino.cc/.

[3] S. Dimitrov. AudioArduino homepage. WWW: http://imi.aau.dk/~sd/phd/index.php?title=AudioArduino.

[4] S. Dimitrov. Extending the soundcard for use with generic DC sensors. In *NIME++ 2010: Proceedings of the International Conference on New Instruments for Musical Expression*, pages 303–308, 2010.

[5] S. Dimitrov and S. Serafin. A simple practical approach to a wireless data acquisition board. In *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 184–187. IRCAM-Centre Pompidou, 2006.

[6] S. Dimitrov and S. Serafin. An analog I/O interface board for Audio Arduino open soundcard system. In *Proceedings of the 2011 Sound and Music Computing Conference*, 2011.

[7] S. Dimitrov and S. Serafin. Minivosc - a minimal virtual oscillator driver for ALSA (Advanced Linux Sound Architecture). In *Not published*, 2011.

[8] U. Drepper. What every programmer should know about memory. 2007. http://people.redhat.com/drepper/cpumemory.pdf.

[9] L. Fried. ladyada.net Wave Shield - Audio Shield for Arduino. WWW: http://www.ladyada.net/make/waveshield, Accessed: 29 Dec, 2010.

[10] M. C. W. Høyerby, M. A. E. Andersen, D. R. Andersen, and L. Petersen. High bandwidth automotive power supply for low-cost pwm audio amplifiers. In *NORPIE2004*, Trondheim, 2004.

[11] G. Kroah-Hartman, B. Ryder, and K. Ober. drivers/usb/serial/ftdi_sio.c. WWW: http://git.kernel.org/?p=linux/kernel/git/stable/linux-2.6.32.y.git;a=blob;f=drivers/usb/serial/ftdi_sio.c, Accessed: 29 Dec, 2010.

[12] M. Nawrath. Arduino Realtime Audio Processing. WWW: http://interface.khm.de/index.php/lab/experiments/arduino-realtime-audio-processing/, Accessed: 29 Dec, 2010.

[13] L. Turchet, R. Nordahl, S. Serafin, A. Berrezag, S. Dimitrov, and V. Hayward. *Audio-haptic physically based simulation of walking sounds*, pages 269–273. IEEE Press, 2010.

[14] S. Wilson, M. Gurevich, B. Verplank, and P. Stang. Microcontrollers in music HCI instruction: reflections on our switch to the Atmel AVR platform. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 24–29. Citeseer, 2003.

[15] www.arduino.cc. Arduino Forum - Measuring Arduino's FT232 throughput rate ? WWW: http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1281611592/0, Accessed: 29 Dec, 2010.

[16] www.atmel.com. Atmel ATmega48A/48PA/88A/88PA/168A/168PA/328/328P datasheet. WWW: http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf, Accessed: 29 Dec, 2010.

[17] www.ftdichip.com. FT232R USB UART IC Datasheet Version 2.07. WWW: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf, Accessed: 29 Dec, 2010.

[18] www.ftdichip.com. FTDI Homepage. WWW: http://www.ftdichip.com/, Accessed: 29 Dec, 2010.

[19] www.usb.org. USB.org - Documents [Specifications home]. WWW: http://www.usb.org/developers/docs/, Accessed: 29 Dec, 2010.

Musical Control of a Pipe Based on Acoustic Resonance

Seunghun Kim, Woon Seung Yeo
Audio & Interactive Media Lab
Graduate School of Culture Technology, KAIST
291 Daehak-ro, Yuseong-gu, Daejeon,
Republic of Korea
seunghun.kim@kaist.ac.kr, woon@kaist.edu

ABSTRACT

In this paper, we introduce a pipe interface that recognizes touch on tone holes by the resonances in the pipe instead of a touch sensor. This work was based on the acoustic principles of woodwind instruments without complex sensors and electronic circuits to develop a simple and durable interface. The measured signals were analyzed to show that different fingerings generate various sounds. The audible resonance signal in the pipe interface can be used as a sonic event for musical expression by itself and also as an input parameter for mapping different sounds.

Keywords

resonance, mapping, pipe

1. INTRODUCTION

Electronic wind controllers have been designed to show the high expressiveness through the generation of various sounds. Fingering information is used in most of the controllers. However, breathing, which is an essential technique for traditional wind instruments, is an optional input parameter. Epipe[1] is an electronic woodwind interface that synthesizes sounds based on tone hole coverage information. The T-Stick[2] has the pipe like shape of a woodwind instrument. A performer can hold, shake, and turn the interface without breath. In The Pipe[3], breath pressure was considered to be a control input for musical expression by using a pressure sensor.

These digital wind controllers detect touch by a number of sensors such as force sensing resistor (FSR). Thus, a complex electronic circuit that controls the sensors needs to be placed in the controller. This makes the controller fragile. Moreover, the many sensors needed result in high development costs and always requires a computer to process the signals from the sensors.

To reduce cost and complexity, we proposed a musical interface that recognizes touches on a pipe by measuring resonance. The original purpose of this work was to develop an interface that creates sonic events by hanging clothes on a clothes airer. However, this paper only focuses on the detection of touches on a pipe without sensors.

This approach is theoretically based on Smyth's design of a musical input device based on the resonances of tuned parallel pipes[4]. In her design, a hole was pierced at one end of the pipe and a microphone was placed in the middle to determine whether the hole was closed or not.

Since the purpose of our work was to recognize the touch/fingering pattern on multiple holes on a pipe, several holes were drilled along the pipe and two microphones were placed, one on each end of the pipe, to measure variations of the resonance.

The biggest advantage of this design is that only microphones are used. Since no sensors are used, the interface is a simple, durable, and not attached to electronics. Additionally, the signal from the microphone can be easily used on a computer as discussed in [4].

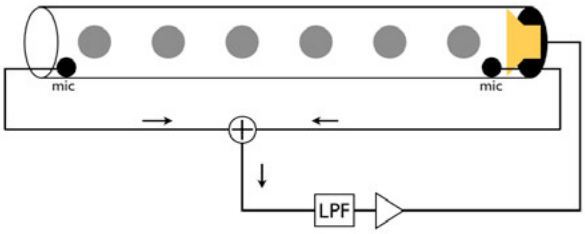


Figure 1: Design of the interface in which the measured sound signal is low-pass filtered, amplified, and played by a speaker unit to amplify the resonance in the pipe

2. DESIGN

Figure 1 shows the design concept of the interface. It included two microphones installed at both ends of the pipe. Signals from the microphones were mixed, low-pass filtered, and amplified. The reason for placing two microphones, one on each end of the pipe, was that the coverage of a hole on the opposite side of a microphone does not significantly change the resonance measured by the microphone. A low-pass filter was used to avoid unwanted excessive howling. The amplified signal was played through a speaker unit placed at one end of the pipe.

Based on the design concept, the interface was developed as shown in Fig. 2. On a PVC pipe of two meters in length, six tone holes (3 centimeters in diameter) were made along the pipe at 30 centimeter intervals. On each ends of the pipe, two pin microphones were installed. The signals from the two microphones were transmitted to a Mackie Onyx 1220 mixer, low-pass filtered by a mixer, amplified by a Pioneer M-1500 power amplifier, and played by a small speaker unit in the pipe.

Figure 2: Prototype of the interface

3. MEASUREMENT

When we started to operate the interface, we could hear the resonant sound. This sound changed as different tone holes were closed and continuously played. However, the generated sound was the same when closing the same tone holes. To demonstrate how the resonance signal is changed when closing different tone holes, the signals were measured and analyzed. Sounds from various tone hole fingering patterns were recorded (one second per each pattern) and analyzed by the fast Fourier transform (FFT). Figure 3 shows the magnitude spectrum when no holes were closed. In the figure, the lowest partial of 311Hz exists and multiples of the frequency were added to the partials of 690Hz and 622Hz. In particular, the partial of 1623Hz, which is the addition of the partial of 690Hz and the third multiple of the lowest frequency, is apparent in the graph, which has a magnitude of -20dB.

Figure 4 shows the magnitude spectrums while each hole was closed. Some signals that were recorded while the third, fourth, fifth, or sixth hole from the right side was closed (Figs. 4c, 4d, 4e, and 4f) have different timbres compared with the signal that was recorded when no holes were closed (Fig. 3). When comparing the magnitudes of the peaks around 311Hz, 1623Hz, and 4850Hz, the differences became apparent (Table 1). Other signals had partials of different frequencies. When closing the first hole, the one closest to the speaker unit, the lowest frequency was 265Hz (Fig. 4a). When closing the second hole, the lowest frequency was 327Hz (Fig. 4b).

Using these characteristics, a simple algorithm can be implemented to classify which hole is closed. At first, the frequency of the highest peak needs to be estimated from the magnitude spectrum. If the frequency is about 265Hz or 327Hz, the first or second hole is closed. If it is about 311Hz, then the second high peak needs to be estimated. If the magnitude of the peak is larger than -20dB, then the fourth hole is closed when the magnitude of the peak around 4850Hz is larger than -40dB and the fifth hole is closed when the magnitude of the peak around 4850Hz is less than -40dB. If the magnitude of the second peak is less than -20dB, then it is regarded that no hole is closed when the magnitude of the peak around 4850Hz is larger than -60dB, the sixth hole is closed when it is less than -90dB, and the third peak is closed when it is between -60dB and -90dB. This algorithm can be implemented using real-time FFT spectrum analyzer software.

Data was also gathered for more than one hole closed. When closing more than one hole, the signals showed clear distinctions in timbre and partial frequencies, as shown in Fig. 5. These sound samples are available at **http://aimlab.kaist.ac.kr/∼asuramk88/pipe**.

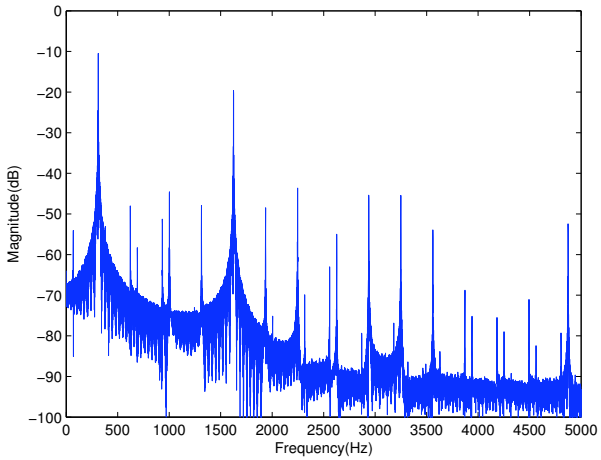


Figure 3: Magnitude spectrum of the resonance in the pipe when no hole is covered

4. APPLICATIONS

A characteristic of this work is that the users can hear sound without any mapping strategies in a computer. Moreover, the sound changes when closing the tone holes in different patterns. Thus, this work can be a musical interface by itself.

The original purpose of this work was to develop a musical interface by detecting clothes hung on the pipe interface. Based on the design, the interface was developed and exhibited in Incheon Digital Art Festival (INDAF), as shown in Fig. 6. It consisted of four pipe interfaces that worked as an interactive clothes airer installation. We used only one microphone in each pipe because the clothes covered more than two holes and changed the sounds audibly with only a single microphone whenever the closed holes were changed. For the audience to hear the resonant sounds clearly, four loudspeakers were also installed in front of the interface.

Information about the resonant sounds may be used for the source of mapping strategies to generate different sounds. The magnitude of each partial can be used as a parameter in how the holes are closed.

5. CONCLUSION

This work presents a musical interface based on the acoustic principle of woodwind instruments. Tone hole coverage of the pipe creates a resonance, and this system amplifies the resonance to generate audible sounds through a speaker unit in the pipe. The sound changes depending on how the tone holes are closed, so this system can be used as a musical interface by itself without any mapping algorithm.

While convenient and cheaper than using multiple sen-

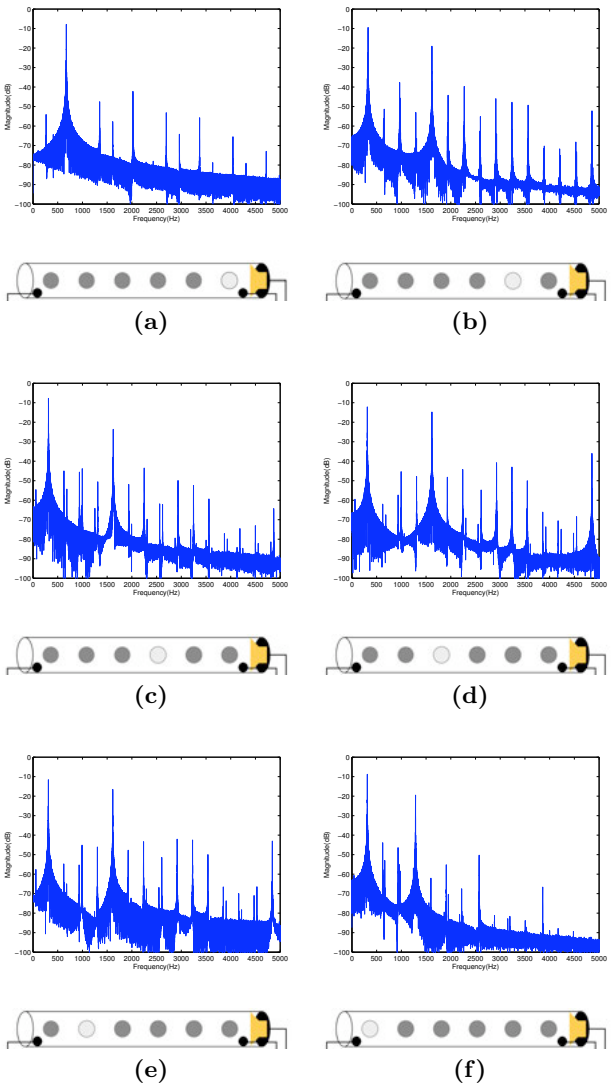


Figure 4: Magnitude spectrums of the signals that close each hole from right to left side

	311Hz	1623Hz	4850Hz
Fig 3	-10.48	-20.36	-52.42
Fig 4c	-7.78	-23.67	-64.29
Fig 4d	-12.14	-14.78	-36.02
Fig 4e	-11.61	-16.53	-43.03
Fig 4f	-8.81	-62.08	-93.22

Table 1: Comparison of dB magnitude of the peaks around 311Hz, 1623Hz, and 4850Hz among the signals having different timbre

sors, this interface has some limitations. One limitation is that the sound plays continuously without manual control of the amplifier. Thus, a computer is required to control the sound volume automatically using software. In addition, the mathematical principle of this work was not explained in full. Smyth[4] presented basic features, but we did not investigate phenomenon in depth once the resonant sound became more complex with two microphones. Further research is needed to determine the mathematics involved.

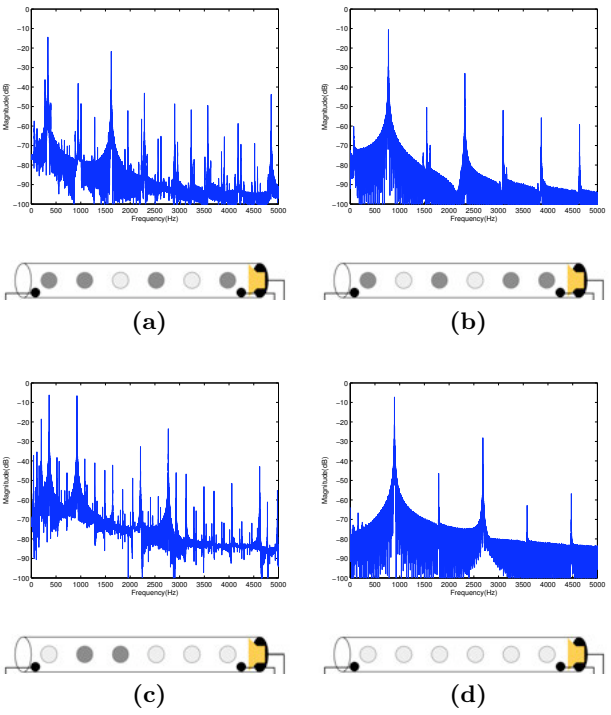


Figure 5: Magnitude spectrums of the signals that close the (a) second and fourth holes, (b) third and fifth holes, (c) first, second, third, and sixth holes, (d) all holes

Figure 6: The interface exhibited in Incheon Digital Art Festrival (INDAF)

6. REFERENCES

[1] S. Hughes, C. Cannon, and S. Ó. Modhráin. Epipe : A novel electronic woodwind controller. In *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME-04)*, pages 199–200, Hamamatsu, Japan, 2004.

[2] J. Malloch and M. M. Wanderley. The T-Stick: From musical interface to musical instrument. In *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME-07)*, pages 66–69, New York, NY, USA, 2007.

[3] G. P. Scavone. The Pipe: Explorations with breath control. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, pages 15–18, Montreal, Canada, 2003.

[4] T. Smyth. Handheld acoustic filter bank for musical control. In *Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME-06)*, pages 314–317, Paris, France, 2006.

Play Fluency in Music Improvisation Games for Novices

Anne-Marie Skriver Hansen
Department of Architecture, Design and Media Technology
Aalborg University, Denmark
email: amhansen@create.aau.dk

Hans Jørgen Andersen
Department of Architecture, Design and Media Technology
Aalborg University, Denmark
email: hja@create.aau.dk

Pirkko Raudaskoski
Department of Communication and Psychology, Aalborg University, Denmark
email: pirkko@hum.aau.dk

ABSTRACT
In this paper a collaborative music game for two pen tablets is studied in order to see how two people with no professional music background negotiated musical improvisation. In an initial study of what it is that constitutes *play fluency* in improvisation, a music game has been designed and evaluated through video analysis: A qualitative view of mutual action describes the social context of music improvisation: how two people with speech, laughter, gestures, postures and pauses negotiate individual and joint action. The objective behind the design of the game application was to support players in some aspects of their mutual play. Results show that even though players activated additional sound feedback as a result of their mutual play, players also engaged in forms of mutual play that the game engine did not account for. These ways of mutual play are descibed further along with some suggestions for how to direct future designs of collaborative music improvisation games towards ways of mutual play.

Keywords
Collaborative interfaces, improvisation, interactive music games, social interaction, play, novice.

1. INTRODUCTION
With interfaces such as the iphone®, the Nintendo Wii® controller, X-box Kinect® there is a potential that music consumption can evolve from being a relatively passive activity to being an active social and expressive activity. The actual musical content can be influenced by the way that people engage with musical expression through a variety of music oriented software and hardware interfaces. Rock Band® and Guitar Hero® are examples of music based game applications where players can engage in music performance, however on a *theatrical* level that does not involve co-creation of improvised music. By theatrical, we mean that players engage with precomposed music through avatars. However, there are several examples of collaborative music interfaces that involve more *dramatic* ways of engaging with music performance: Blaine, Fels and Weinberg have discussed mapping of joint user action in networked interfaces [1][16].
Many collaborative music applications also take advantage of commercial interfaces like the iphone that have built-in sensor capabilities and can be added to a network. Some examples are presented in [10][15][11][12]. These kinds of collaborative music interfaces and interface applications could define a new kind of "casual games", where the auditive, and not the visual is in focus, and where the joy of play replaces the idea of a 'high score' [6].
This paper discusses the role of a music game application and how it encourages players to 1) establish mutual awareness towards each other's actions and joint attention towards the object of music creation, 2) engage in varied forms of individual and mutual expression and 3) engage in *play fluency*. By the term *play fluency* is meant meaningful musical expression perceived by players and a potential audience. Play fluency could be a sign of flow and the music game could potentially be intrinsically rewarding because it inspires players to engage in *autotelic* activity [2]. In the Continuator interface, Pachet has investigated how a player engages in a flow experience while improvising together with a music application as a 'co-player' in turn taking sequences [9]. This paper presents how two players improvised together when their musical performance was triangulated with a music application – if and how the music application facilitated play fluency. The main objective of this study was to see if a music-based game application that captured limited and specific aspects of mutual play was able to give appropriate sound feedback when two players managed to establish play fluency together.

2. GAME DESIGN
The music game was programmed in Max Msp [8] for two Wacom Intuos4® pen tablets [14]. This kind of interface has previously been used as an expressive electronic music instrument [18]. The game application borrowed the idea of drawing in order to make it easy to play for novices: Two players could either draw dots and lines, scratch movements and circles, where the sound feedback would differ according to these draw styles (see the sections 2.1-2). Players would get additional sound feedback if they chose to draw the same draw style and were able to synchronize and time their movements with each other. The game design was inspired from the idea of musical grounding, a concept presented in the field of music therapy that describes how a music therapist can establish a sort of musical and communicative understanding of a client by for example reflecting aspects of the client's play style [17]. Here, the game application was designed to notice only limited aspects of two players' mutual grounding: choice of playstyle synchronization and timing.

2.1 Sound Feedback: Individual Action
Measured x and y pen positions were translated into simple draw styles: dots/lines, scratch movements and circles (see figure 1-3). One player's individual string instrument sound was based around high frequencies, and the other player's individual string instrument sound was based around the low frequencies. All frequencies were fixed around a Balinese Pelog scale. It did not matter where on the tablet the draw styles (dot/line, scratch movements and circles) were made. Some features connected to the draw styles were detected: Size and

speed, line and scratch degree (360) and circle drawing direction (cw/ccw). The shape detection was relatively simple, so the application could detect the draw movements in real-time (within a sample rate of 20 to 200 milliseconds). The extra features determined which combinations of tones were activated. In addition the pen's x and y tilt angle influenced the length and volume of each activated tone.

2.2 Sound Feedback: Mutual Action

When two players chose to use the same draw styles (dots/lines, scratch movements or circles) in pairs, and when they drew at the same speed, they would activate an additional sound layer: Piano chords were played back on top of each individual player's sounds. The rhythm of the piano chords was at the same pace as the mutual pace of the two players. If players kept drawing at the same speed, the rhythm structure of the piano would elaborate around the rhythm that the two players had found together. If the offset time between scratch peak points and circle top points was low, the two players activated high pitch chime sounds.

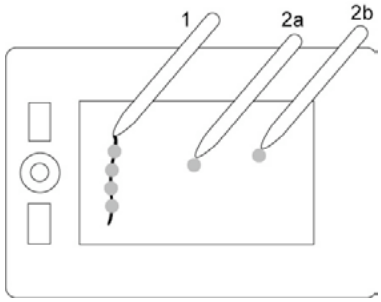


Figure 1: Dots and lines. The grey dots = tones activated along a line (pen1), or tones activated when the pen touched the tablet (2a and 2b).

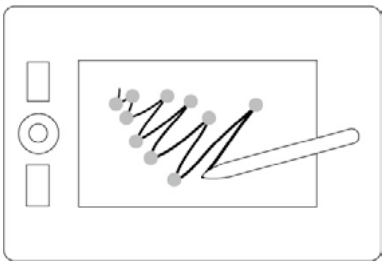


Figure 2: Scratch movement. Grey dots = tones activated at the points of direction change. Scratch area and degree in 360° was also noticed by the game engine.

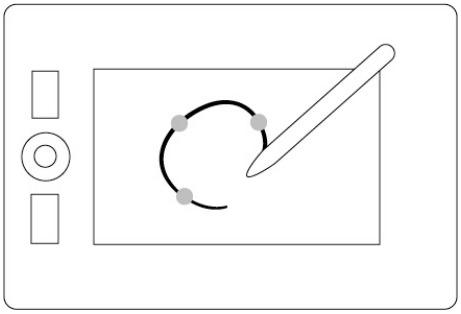


Figure 3: Circle movement. Grey dots = tones activated along the curved line of the circle. Circle area and clockwise /counter-clockwise movement was also noticed.

3. EXPERIMENT PROCEDURE

In nine game sessions we documented how two players played together. The teams consisted of either two females (4 teams total) or two males (5 teams total). Documentation happened in two ways: A video camera filmed the two players from the side, while the game application logged all incoming pen data and metadata generated by the game application. (see figure 4).

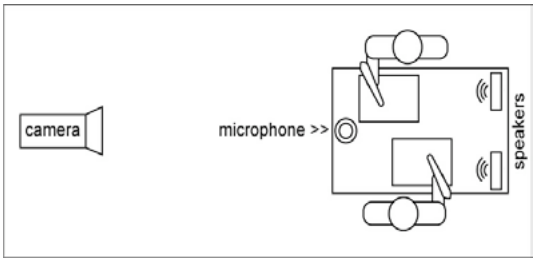


Figure 4: Two players sat at a table opposite each other with the pen tablets in front of them. A microphone was placed on the table to record what the two players said. Speakers next to each player played individual sound feedback. The sound feedback that happened as a result of joint action was centered or panned between the two speakers.

When the experimenter introduced the game and the different draw styles, the players were asked to find 'additional sounds'. They were told to collaborate, but not what to do in order to find the additional sounds. When each player had explored his/her individual sounds, the two players could play together as long as they wished.

4. DOCUMENTATION AND MEASUREMENT OF USER ACTION

The video documentation of all game sessions provided material for qualitative analysis of individual and mutual player action and the social context that surrounded this action. In order to see how players negotiated mutual play and engaged in sequences of play fluency with each other, we used conversation analysis - see [3][4][5][7][13]. The video was analyzed on several levels: Did players for example use utterances, gestures and gaze to negotiate play? Was there a special rhythm or hierarchy of actions? Could individual and mutual pauses be related to the way players understood the sound feedback, especially the additional sound feedback that happened as a result of mirrored play style and common speed and timing?

5. RESULTS

This section presents first some video examples of characteristics of play fluency that we found in the nine game sessions. In general, it was impossible to find a single way in which two players engaged in play fluency together. Players were very inventive, and each player team acted differently. What all player teams had in common was the process of establishing play fluency:

First, in order to find a shared focus and start a musical relationship, players either mirrored each other by doing the same (getting the additional sound feedback), or one player tended to accompany the other player (see appendix, video 2a, 3b, 6d, 7a-c, 8a, 8h, 9a-e). At most times, the player who initiated a draw style ended up playing solo, while the other player entered their relationship through accompaniment (see appendix, video 1a, 1d, 4a, 5a). However, in session 6 and 8, the player who introduced a relationship, introduced a background for a possible solo (see appendix, video 6a-c, 8g).

Second, there were many different kinds of relationships in terms of draw style. No matter which combination of draw style players engaged with, players tended to explore a variation of a found relationship shortly after it was clear that they had

established a relationship (see appendix, video 3a, 3c, 8a, 8b, 8e, 8d, 9b, 9c). In one case (see appendix, video 1f), players changed the tempo as a way of varying a found relationship. It seemed like there was a hierarchy in that players needed to first agree on draw style combination before they started to engage in an exploration of e.g. pen position, pen tilt and play speed.

Third, players repeated each other's utterances in a turn taking relationship (see appendix, video 1c, 1e, 4b, 8f, 8h). In the case of 1e, players ended up sharing the same timing, whereafter they started to play different play styles simultaneously, exploring other ways of playing together. In 4b it is clear that the additional sound feedback did not support turn taking.

Fourth, players tried to make sense of the additional sound feedback that sometimes happened as a result of their mutual play. Perhaps this was because players acknowledged that their task 'find additional sounds' was done. Some players also looked at the computer screen in order to find an answer. Some players ignored the additional sound feedback.

Fifth, play fluency seemed to arise, when two players managed to stay focussed on very limited ways of expression, often repeating a sequence of tones with slight variations (see appendix, video 1d). In a few cases, players negotiated a play relationship verbally (see appendix, video 9a-e). Some considered very sophisticated relationships that regarded the graphical layout of pen actions (see apendix, video 3e-f and 8e and figure 5).

Video 3f:

Right player: "Try to drive around, then I put dots around."

Video 8e:

"What now if we keep moving across a specific point,
for example here, and then you run across that?"

Figure 5: Transcription of player utterances in video 3f and 8e (see appendix).

6. ANALYSIS

In general, the two players shifted between individual exploration and joint expression. The types of player engagement shifted between the three types of engagement presented by Ben Swift et al.: individual, unilateral and bilateral engagement [3]. In some cases, while one player engaged in individual exploration, the other player followed along without the first player was aware of it. In other cases both players were mutually aware of each other's actions. The mutual player awareness (or problems in finding it) was visible in the following types of communication:

6.1 Talk and Utterances

None of the teams talked very much while exploring joint improvisation. Usually smiles and laughter was used to indicate if players had found a shared form of expression. They also sometimes commented on the sound feedback with single words like "hmm" and "ah". When a few teams did talk, it was because they needed to negotiate some very specific pen actions with each other, and here some players used deictic gestures to explain.

6.2 Gestures and Postures

None of the teams used gestures that were related to musical expression. This was perhaps because they, unlike trained musicians, had no formal gesture vocabulary to use. However, when a player wanted to be very explicit about his/her actions, s/he tended to lift the pen higher than usual. This also happened at the end of each phrase that a player introduced. Players used body postures to direct each other's pen movements (this is very clear in video 9a-e, see appendix). Players tended to move their bodies more, when they were engaged in play fluency. In the two examples, the player who introduced the leading musical content was very explicit about what s/he did. In 1d Right moved his torso along with the arm movements when he scratched and drew circles. In 4a Right introduced a melody by nodding along with the first couple of tones.

6.3 Gaze

In general, female teams tended to exchange gaze more than male teams. Players often switched between looking at their own tablet and the other player's pen and tablet. In the following two examples gaze patterns in successful play fluency sections from a male and a female team, are covered in order to understand how play fluency was negotiated: In section 1d Right did not look at Left before towards the end of the found relationship. This was in order to indicate a desire to 'take the floor' by coming up with new material. In section 4a the two players looked at each other in turns. This could be to check if the other player was following along, and if the player who guided their mutual play had noticed that the other player was following. In both sections, gaze and pauses were intricately connected: In 1d Left looked at Right's pen and tablet a moment before Right introduced the first phrase - perhaps in order to get an idea of timing. Perhaps it was easier for Left to follow Right's movements, because Right is left-handed? In general, Left checked more with gaze what Right was doing than vice versa. When Right then looked at Left's pen and tablet before phrase 3 where Right introduced circles, Right's gaze was a guidance. Then Left checked what Right did, when he actually switched to drawing circles. Left's phrases 3 through 5 could be interpreted as one long phrase that was an elaboration on phrase 1 and 2. In the entire video clip, the game application did not provide any aditional sound feedback. The game application was not designed to interpret this type of play relationship as 'meaningful'. In 4a both players started to look at each other's pen and tablets in order to find a common relationship together. Right looked at Left when introducing the first tones of a melody, while Left responded by looking at Right's pen and tablet while smiling. When Right doubled the tempo she looked at Left when she realized that Left followed her quite well.

7. DISCUSSION AND FUTURE WORK

This paper has presented a qualitative evaluation of how a music based game application supported players in establishing play fluency. On one level, the game application did successfully support players in improvising together. By providing players, who were not trained musicians, with a recognizeable physical interface and two kinds of string instrument sounds, there was enough material that players could use to relate to each other with. It was easy for the players to understand the three draw styles, and most players intentionally used combinations of those. However, the game application did not succeed in triangulating the two players mutual play. Very often players did not understand the additional sound feedback that happened when players used the same drawstyles and played those at the same speed and timing. Although the game application could measure the combinations of different draw styles, and all the features of the pen movement connected to these draw styles, only a fraction of these individual and joint interaction data were mapped to sound output. It was very clear that players expected more sound feedback as a result of even small changes in their mutual actions. The game application had too many expression

possibilities: Players could combine the different draw styles, and they could vary them with pen tilt, pen position, drawing direction and size. The variations in mutual draw styles were so big that no single description of how they played together was sufficient. It would have been a big task to design a game that could provide sound feedback on all the three kinds of draw styles and their related features. Instead, in future music based games, we suggest to narrow down the expression possibilities, so that it is possible to map all play combinations and features to some sort of musical and/or sound effect output.

7.1 Play Fluency in Joint Improvisation

As most of the video examples show, play fluency happened when players focussed on a few ways of expressing themselves with the available sounds. One player's focus and repetitive movement gave the other player a chance to grasp what was going on and try out ways of attuning his/her actions to the first player's actions. A game application that only asks players to draw lines could afford more focus on how players draw lines with each other. This would also focus the two players attention towards varying a found relationship even more, because a game application could support all types of line drawing in the sound feedback. The results and the analysis of the game application presented in this paper offered a glimpse into a wide variety of ways in which players chose to establish play fluency. Future designs could elaborate on a selected set of means of expression that were logical to players while engaging in play fluency together.

7.2 The Role of the Game Application

It was clear that the idea of musical grounding that was implemented in the game application was too narrow. It did not embrace the wide variety in which players established musical grounding through all the available expression possibilities. The idea of a triangulation of two players mutual play should be re-evaluated according to what a game application in fact can measure out of the entire embodied interaction of the social act of musical improvisation. It was seen that there was a hierarchy in how players explored the draw styles and draw features. A game application could be designed to give and vary sound feedback according to how: 1) draw style combinations are chosen, 2) variations of draw styles are made and 3) mutual timing and speed is negotiated among players.

8. ACKNOWLEDGMENTS

Thanks to students at Aalborg University for participating in game sessions. Also, special thanks to students from the music therapy department at Aalborg University for providing critique of the game design.

9. REFERENCES

[1] Blaine, T. and Fels, T. Contexts of Collaborative Musical Experiences. In *Proceedings of the Conference of New Interfaces of Musical Expression (NIME'03)* (Montreal, Canada, May 22-24, 2003), ACM Press, 2003, 27-33.

[2] Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience.* Harper & Row Publishers Inc. 1990.

[3] Goodwin, C. Conversation Analysis. In *Annual Review of Anthropology*, Annual Reviews, 1990, 283-307.

[4] Goodwin, C. Action and Embodiment within Situated Human Interaction. In *Journal of Pragmatics, no. 32*, Elsevier, 2000, 1489-1522.

[5] Goodwin, C. Participation, Stance and Affect in Organization of Activities. In *Discourse & Society, no. 18.53*, SAGE Publications, 2007, 53-73.

[6] Juul, J. *A Casual Revolution: Reinventing Video Games and Their Players.* MIT Press. 2010.

[7] Kendon, A. Movement Coordination in Social Interaction. In *Conducting Interaction: Patterns of Behavior in Focussed Encounters*, Kendon, A. (ed.), Cambridge University Press, New York, 1990, 91-116.

[8] Max Msp: http://cycling74.com/products/maxmspjitter/

[9] Pachet, F. On the Design of a Musical Flow Machine. In *A Learning Zone of One's Own*, (Tokoro and Steels eds.), IOS Press (Amsterdam, The Netherlands), 2004, 111-134.

[10] Swift, B. et al. Engagement Networks in Social Music-making. In *Proceedings of OZCHI* (Brisbane, Australia, November 22-26, 2010), ACM Press, 2010.

[11] Tahiroglu, K. Towards and Experimental Platform for Collaborative Music Performance. In *Proceedings of Sound and Music Computing Conference (SMC 2009)* (Porto Portugal, July 23-25, 2009), 2009, 183-188.

[12] Tanaka et al. Facilitating Collective Musical Creativity. In *MULTIMEDIA (MM 2005)* (Singapore, Malaysia, November 6-11, 2005), ACM Press, New York, 2005, 191-198.

[13] Tannen, D. Silence: Anything But. In *Perspectives on Silence*, Tannen, D. and Troike, M.S. (eds.), Ablex Publishing Corporation, New Jersey, 1985, 93-112.

[14] Wacom Intuos4: http://www.wacom.com/intuos/

[15] Wang, G. et al. SMULE = Sonic Media: An Intersection of the Mobile, Musical, and Social. In *Proceedings of the International Computer Music Conference (ICMC 2009)* (Montreal, Canada, August 16-21, 2009), 2009, 283-286.

[16] Weinberg, G. Interconnected Musical Networks: Toward a Theoretical Framework. In *Computer Music Journal, vol. 29, No. 2*, MIT Press, 2005, 23-29.

[17] Wigram, T. *Improvisation. Methods and Techniques for Music Therapy Cliniciand, Educators and Students.* Jessica Kingsley Publishers, London, UK, 2004.

[18] Zbyszyński, M. et al. Ten Years of Tablet Musical Interfaces at CNMAT. In *Proceedings of the Conference of New Interfaces of Musical Expression (NIME'07)* (New York, USA, June 6-10, 2007), 2007, 100-105.

10. Appendices may follow the references

The following nine web addresses are links to selected video sequences of the nine game sessions where 18 persons participated. In order to see the videos, this following password is needed: AMSH5research. The selected videos show sequences where the teams established mutual play fluency. The sub-section times are indicated on the website below the video.

Video 1a-f: http://vimeo.com/19119476
Video 2a-b: http://vimeo.com/19119262
Video 3a-g: http://vimeo.com/19119134
Video 4a-b: http://vimeo.com/19118761
Video 5a-c: http://vimeo.com/19118652
Video 6a-e: http://vimeo.com/19118358
Video 7a-b: http://vimeo.com/19117874
Video 8a-i: http://vimeo.com/19117700
Video 9a-e: http://vimeo.com/19116889

On the following two links pen x and y positions for the examples 1d and 4a are presented as time/color diagrams.
Pen positions I: 1d/left player: http://vimeo.com/19384019
Pen positions II: 1d/right player: http://vimeo.com/19384067
Pen positions III: 4a/left player: http://vimeo.com/19384161
Pen positions IV: 4a/right player: http://vimeo.com/19384216

The Bass Sleeve: A Real-time Multimedia Gestural Controller for Augmented Electric Bass Performance

Izzi Ramkissoon
New York University
34 West Fourth St.
New York, N.Y. 10012
IzzRk@aol.com

ABSTRACT
The Bass Sleeve uses an Arduino board with a combination of buttons, switches, flex sensors, force sensing resistors, and an accelerometer to map the ancillary movements of a performer to sampling, real-time audio and video processing including pitch shifting, delay, low pass filtering, and onscreen video movement. The device was created to augment the existing functions of the electric bass and explore the use of ancillary gestures to control the laptop in a live performance. In this research it was found that incorporating ancillary gestures into a live performance could be useful when controlling the parameters of audio processing, sound synthesis and video manipulation. These ancillary motions can be a practical solution to gestural multitasking allowing independent control of computer music parameters while performing with the electric bass. The process of performing with the Bass Sleeve resulted in a greater amount of laptop control, an increase in the amount of expressiveness using the electric bass in combination with the laptop, and an improvement in the interactivity on both the electric bass and laptop during a live performance. The design uses various gesture-to-sound mapping strategies to accomplish a compositional task during an electro acoustic multimedia musical performance piece.

Keywords
Interactive Music, Interactive Performance Systems, Gesture Controllers, Augmented Instruments, Electric Bass, Video Tracking

1. INTRODUCTION
In the past, when working on developing a performance setup to combine both live electronics and a string instrument the focus has been on extending traditional technique by using gesture acquisition based on existing relationships between the performer and string instrument. The use of existing gestures, in regard to the common practice of string instrument technique, has led designers to use instrumental gestures to control electronic sound. It is important to maintain the technique that many performers have developed over years of practice, while still being able to control the electronic aspect of the performance. This method to instrument augmentation has led to many systems in violin, cello, acoustic bass performance.

2. THE BASS SLEEVE
The design of the Bass Sleeve allows for a performer to control expressive parameters of a sound using motions other than hand to string relationships. In the design, ancillary gestures such as forward and backward foot motions, knee movement, knee bending, and quick hand gestures were used to extend the range of expressive gestures within the performance of an electronic electric bass performance. In the context of a performance, these motions create metaphors of tension and release that shape the sound viscerally, communicating qualities of effort to the audience, while extending the range of expressive gestures. The design of the hardware interface augments the gestures of the electric bass and increases the independent control over sound processing creating a variety of new gestural relationships for bassist.

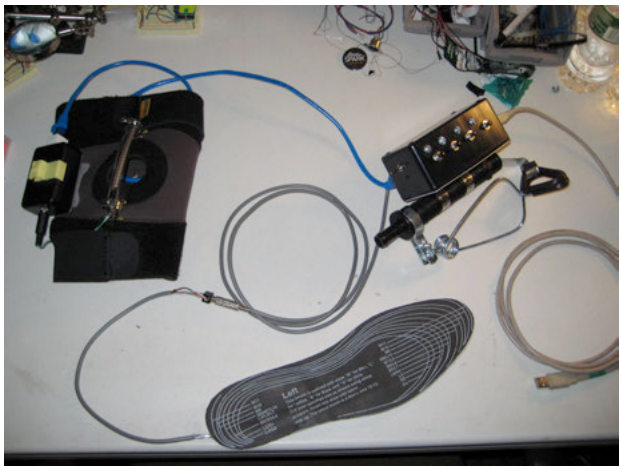


Figure 1: Bass sleeve full system prototype including the knee controller, foot controller and control box.

3. HARDWARE DESIGN
The Bass Sleeve incorporates both a knee mounted controller and an insole for foot control. Both of these parts of the controller use ancillary movements to shape the sound in a real-time electric bass performance. The movements that were selected are outside the range of effective gestures when controlling the sound of the instrument. In a situation where an electric bassist performs with a laptop it becomes difficult to continuously control both the laptop and the bass with only two hands. This might seem a bit obvious but many times a controller is made to interrupt an expressive performance with additional gestures that have nothing to do with an expressive performance. The gestures introduced into the performance control something other than the actual sound of the instrument creating additional gestures that lower the communication of expression during a concert. A solution to incorporating controls and maintaining the level expressivity in a performance can be seen with the addition of ancillary gestures on pre-existing traditional gestures. This approach is an

alternative to using traditional technique as gestural controls and is explored with both the knee and foot controller.

3.1 The Knee Controller

Figure 2: The knee controller uses a bend sensor and accelerometer.

The knee controller is made using a knee brace, a flex sensor, accelerometer, safety pins, clear tubing, Velcro, Ethernet port, small black project box and clamps. The flex sensor is inserted into a plastic tube to act as a buffer. This relieves some stress from the bend sensor during a performance. In a previous test the performer would bend the knee rather aggressively sometimes either breaking or overloading the sensor causing loud pops and crackles interrupting the sound with unwanted digital artifacts. With the addition of the plastic tube the stress that occurs with the bending of the knee is dissipated throughout the material allowing the bend sensor to bend freely within the durable tubing. The tube is attached to the knee brace using clamps and safety pins. This makes it easy to replace worn bend sensors while having a durable form. The bend sensor output uses an 8th inch jack making the input modular. The entire system follows the idea of being modular allowing customization for each individual performance or composition. On the side of the knee is a small black box that houses the accelerometer. The bend sensor output is connected at the bottom.

The accelerometer on the side of the knee and is used to map the motions of the knee through space. The placement of the accelerometer allows accurate x, y, and z control. Velcro and a safety pin are used to fasten the box to the side of the knee. In the development of this system it was found that Velcro could be used to affix many sensors, although when performing sweat moisture can create problems with sensor placement. This led to the addition of a safety pin to ensure that there would be no movement of the sensor.

3.2 Insole Foot Controller

The second part of the bass sleeve was the foot controller. The foot controller is made using two force sensing resistors, hot glue, wire cable, port and a shoe insole. The FSR wires are hot glued to the surface of the shoe insole. The hot glue is used to both insulate the wires from sweat and secure them to one place on the insole. The insole has a port to insert the FSR making the sensor easy to replace. The output from the FSR wires is a stereo 8th of an inch jack that can be extended to the control box using a stereo cable. After all the cabling was finished both the left and right insoles were glued together to create a buffer between the foot sensors and shoe.

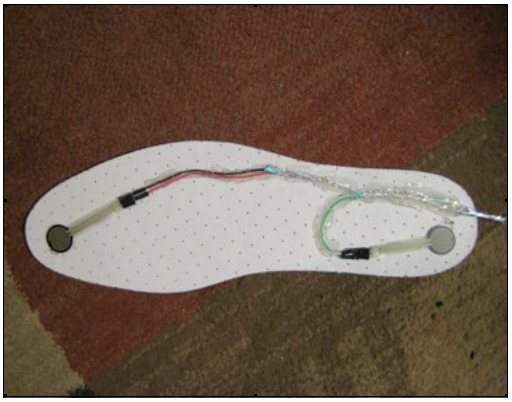


Figure 3: Foot controller using two FSR's.

The foot controller adds the possibilities of sensing the weight distribution of the leg from front to back. This led to the placement of sensors at the front toe and back heel for weight distribution. Also, when in resting position the weight was neither all the way on the toe nor on the heel but evenly dispersed around the foot. This relationship will be explored with various mapping strategies. In tandem both the knee controller and the foot controller acquire the motions of the leg either thrusting forward and down, back and up, side to side or any combination of these motions. Both can be easily detached or attached in any giving performance making it very modular for many different types of mappings.

3.3 The Control Box

The control box was made to attach onto the bottom of the bass visually linking all of the sensors on the leg to the actual bass. The box placement was important for both aesthetic and functional reasons. Having the box in visual site attached to the bass creates the sense of augmenting. In the past, augmentation many times comes with the sensors all over the performance area of the instrument. This was more of a subtle solution to connecting the two worlds. It also serves as part of the controller with buttons for mode selects. This shortens the distance needed to travel to press a key and is smaller than a foot controller because of the form factor of the hands. The box is used in performance that same way any other tone control on your bass would be used only this controls both tone, texture and video.

Figure 4: The control box mounted to an electric bass.

The control box is made using a medium size project box, switches, buttons, plastic pipe fixtures, clamps, Arduino mini, Arduino to USB, Ethernet port, 8th inch stereo jack and a light clamp. Inside the box an Arduino mini microprocessor is used connect to a mini USB adapter. The Arduino mini has 8 analog inputs and 12 digital inputs. The control box has 4 wired

buttons hardwired to the microprocessor with the possibility of expansion to 8, if needed. The controls are used to turn devices on and off, shuffle through modes, and set levels.

3.4 Video Tracking

Elements of video tracking were used in the software patch to gather information about the position of the performer. Many times sensor-based instruments localize parts of the body gathering small movements. The Bass Sleeve incorporates both small and large movements using the body-mounted sensors to capture very small gestures and video tracking to capture larger gestures. When performing multiple tasks in a live performance it is hard to think about performing on an instrument, processing the sound of your instrument and then controlling video real-time. The video-tracking patch creates an algorithmic relationship between the performer's place in space and video processing. The video processing mappings were, the closer a performer gets to the camera the larger the processed visual gets, the greater amount of side-to-side movement produces variations in the horizontal placement of the image.

4. ANCILLARY GESTURE CONTROL

In the Bass Sleeve project ancillary gestures were mapped to sound synthesis and processing. The main goals of this project was to delve into the possibilities for gestures other than traditional playing technique, effective gestures, and accompanist gestures to control sound parameters in a live electric bass performance. The use of ancillary gestures in live performance to control sound has played a minimal part in the creation of modern electronic music, even more so the relationship between expressive gestures and sound. It was found that in a performance the ability to show effort and create tension and release through a gesture was an effective way to communicate expressive musical qualities such as pitch shifting, timbre manipulation, stutter and drone effects. The methods that were used to create the tension and release in the sound were body motions that had the same qualities such as the bending of the knee and foot pressure. The ability to use effective, accompanist and ancillary gestural control shaped the performance into a full body expressive instrument is important when developing an expressive full range controller.

4.1 Ancillary Gesture Mapping Strategies

Table 1. Bass Sleeve Mappings

Front foot FSR – pitch shifter
Back foot FSR – sampling
Knee bend sensor – low-pass filtering
Knee Accelerometer X – synthesis pitch
Knee Accelerometer Y – synthesis LFO
Knee Accelerometer Z – synthesis LFO
Bass Box 1 up – drum volume
Bass Box 2 up – drum phase start
Bass Box 3 up – video tracking enable
Bass Box 4 up – mounted sensors tracking enable
Bass Box 1 down – Bass Sleeve ON/OFF
Bass Box 2 down – delay repeater
Bass Box 3 down – delay feedback
Bass Box 4 down – filter select

These mappings were determined in the initial experimentation of the design. The design of the mappings was based on the relationships between the motions of the body and the compositional process that it would control. The front foot, in the first set of mappings, was made to trigger the sampling of the electric bass and the back heel was mapped to pitch shifting. This was an experiment with the mappings and not the intent of the original design. It was found that the back heel did not communicate as much effort and tension as the front foot did when using pitch shifting. It was also found that the back heel used for pitch shifting created a tiresome performance practice that was difficult to maintain over an hour performance. These mappings were reversed and put back to the original concept, which was to use a downward thrust to signify to the audience and fellow performers that effort was being put into pitch shifting the sample. The downward thrust with pressure on the front foot was effective in communicating this particular sound. The performance gesture that was used shifted pressure from the front of the foot to the back of the foot processing the sample with foot pressure. This process produced variations in sampling and pitch.

One of the mappings used in the programming of the Bass Sleeve was low pass filtering to the amount the knee bent. The choice to use low pass filtering was an important compositional decision. The relationship between the effort used to bend the knee and the sound processing created a metaphor of tension and release within the performance. When experimenting with the knee sensor it was also mapped to pitch shifting which produced a meaningful relationship as well. The relationships that related most to the knee gestures were those that had the similar qualities of tension and release such a speed control, scrubbing, clean to distortion, but in the end low pass filtering had a relevant quality that was desired in the overall signal flow of the Bass Sleeve. Alternative mappings can be used in the knee depending on how a composer wishes to form relationships between the controller and sound.

The knee controller contained an accelerometer. The accelerometer was mapped to sound synthesis controlled by both the knee and the pitch of the bass. The sound synthesis was at the top of the signal flow so that the low pass filtering could process it. The mappings of synthesis to the knee were a novel approach to the controller, which sounded better than it looked.

The mappings used in the control box can be re-mapped depending on the important parameters, functions, events, and mode selects in any given composition or improvisation. A performer can determine beforehand the necessary quick controls and map them to the control box, allowing more control away from the computer and closer to the instrument.

5. SOFTWARE

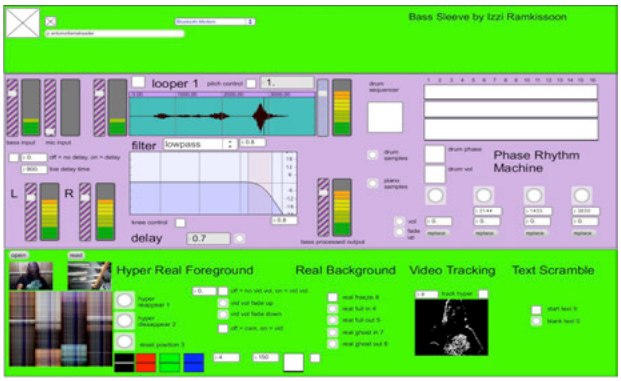


Figure 5: Screenshot of the laptop screen during a practice session with the Bass Sleeve system.

The bass sleeve system was designed for composition and live performance. The software was designed for a specific direction in composition. The programming developed to explore methods to develop a small looping sample with delay, filtering, pitch shifting creating granular clouds and stutter effects. This was in response to the current state of loop-based performances. Many electronic musicians use looping devices to create live electronic music. The ability to loop gives the performer and composer the ability to create layers on layers of musical material. The issues in this type of composition are the process in which composer develop static loops to produce variety. Many time loop compositions take the form of adding and subtracting layers with little manipulation of the sample loop itself. The approach to looping in the Bass Sleeve programming allows a small loop to have life outside of repetition by processing the samples sound over time. The interest in developing the sample using certain techniques came from the exploration of digital manipulation in earlier compositions including pitch shifting, sample speed, filtering and delay. In electronic music the use of repetition can be seen in pieces such as Steve Reich's Electric Counterpoint or Terry Reilly's in C. The use of effects to create repetition has also played an important role in modern music. Much of these ideas were the inspiration for the programming.

6. CONCLUSION

In the development of the Bass Sleeve it was found useful to use ancillary gestures for music control. These extra gestures give independence to one-to-one and one-to-many mapping strategies allowing a musicians complex control away from instrumental gestures. The combination of both traditional electric bass instrumental technique and computer control can be difficult when performing. The Bass Sleeve introduces a novel solution to performing with an electric bass and the laptop by controlling and manipulating processed sound in live performance situations using ancillary gestural control. It was found that small gestures could be used with the hands similar to the motion to turn a volume potentiometer or flip a pickup switch and larger gestures such as side-to-side movement for audio and visual panning. Incorporating ancillary gestures into performance technique allows for a more expressive full body performance based on the increase in interaction a performer can have with an audience, other performers, the laptop and traditional instrument. The goal of this system is to approach the realm of having a seamless relationship between the performer-to- instrument and performer-to-computer, with the relationship being performer-to-overall instrument.

7. FUTURE WORK

In the future version of the Bass Sleeve augmented instrument design some of the hardware issues such as sensors slipping, sensor selection, durability and reliability on the knee gestural controller will be redesigned to increase its performance in live musical applications. The redesign will include an exploration in alternative options to sense the bending of the knee other than a bend sensor, other options to mount the bend sensor to produce a greater amount of durable and reliability, and the possibility of additional programming to compensate for sensor erratic input when a sensor slips or is overloaded.

The application of the knee bending gesture to sound will be explored further since the mappings were specific to the current composition. In the future version, a database of different modes, functions and compositional approaches will be programmed to allow the Bass Sleeve to have a variety of relationship to express during a performance. The database will include different sets of processing that all relate to each other with a modular approach to programming allowing the user to experiment with different signal chains. The system will also have a modular approach to hardware design allowing for additional ancillary gestural hardware attachments.

In addition to the redesign and redefining of the system through mapping, a greater amount of compositional techniques will be explored in the future work with the Bass Sleeve. Some specific techniques that will be explored in the Bass Sleeve will be relationships between audio and video processing and mapping such as live input amplitude mapped to the visual pillars, methods to analyze and synthesize the bass sound creating counterpoint lines using MIDI, the processing of synthesized counter point lines during a performance by a gesture using filtering and LFO, the mapping of independent layers or counterpoint lines to specific ancillary gestures to form relationships to the live electric bass and the programming for real-time improvisation.

The future of the Bass Sleeve system will be for personal use since many of the concepts still need to be tested in a live performance setting. In the redesign of the interactive performance system certain standards of usability must be met before other musicians can use the system. The controller will have to support a very general approach to satisfying the performance needs of any electric bassist before can be introduced as a device for public use.

8. REFERENCES

[1] Bahn, C., Hahn, T., & Trueman, D. "Physicality and Feedback: A focus on the Body in the Performance of Electronic Music.

[2] Hunt, A., Wanderly, M., Kirk, R. "Towards a model for Instrumental Mapping in Expert Musical Interaction" York Music Technology Group and IRCAM.

[3] Hunt, A., Wanderly, M., Paradis, M. "The importance of parameter mapping in electronic instrument design" NIME 2001.

[4] Hunt, A., Ross, K. "Mapping Strategies for Musical Performance" University of York.

[5] Livingston, H. "Paradigms for the new string instrument: digital and materials technology", in Organized Sound Volume 5, Issue 3, Cambridge University Press

[6] Lahdeoja O., Wanderley, M., Malloch, J. "Instrument Augmentation using Ancillary Gestures for Subtle Sonic Effects" SMC 2009.

[7] Lahdeoja O., Wanderley, M., Malloch, J. "Instrument Augmentation using Ancillary Gestures for Subtle Sonic Effects" SMC 2009.

[8] Turchet, L., Serafin, S., Dimitrov, S. Nordahl, R. "Physically Based Sound Synthesis and Control of Footsteps Sounds" Proceedings of the thirteenth International Conference on Digital Audio Effects, Graz, Austria, September 6-10, 2010.

[9] Wanderly, M. "Non-Obvious Performer Gestures in Instrumental Music" IRCAM – Centre Pompidou.

[10] Wanderly, M., Depalle, P., "Gestural Control of Sound Synthesis" IEEE 2004.

[11] Wanderley, M., Vines, B., Middleton, N., McKay, C., Hatch, W. "The Musical Significance of Clarinetists' Ancillary Gestures: An Exploration of the Field." Journal of New Music Research. 2005, Vol. 3

The KarmetiK NotomotoN: A New Breed of Musical Robot for Teaching and Performance

Ajay Kapur Michael Darling Jim Murphy

Jordan Hochenbaum Dimitri Diakopoulos Trimpin

KarmetiK LLC
Reno NV, USA
info@karmetik.com

ABSTRACT

This paper describes the KarmetiK NotomotoN, a new musical robotic system for performance and education. A long time goal of the authors has been to provide users with plug-and-play, highly expressive musical robot system with a high degree of portability. This paper describes the technical details of the NotomotoN, and discusses its use in performance and educational scenarios. Detailed tests performed to optimize technical aspects of the NotomotoN are described to highlight usability and performance specifications for electronic musicians and educators.

Keywords

Musical Robotics, Music Technology, Robotic Performance, NotomotoN, KarmetiK

1. INTRODUCTION

The field of musical robotics has long been in the domain of highly specialized one-of-a-kind instruments. Such instruments are typically only played by their creators, while those unfamiliar with the robotic instrument would likely be faced with a confusing system which is highly specific to the given instrument. With the KarmetiK NotomotoN, we aimed to address these issues by creating a user-friendly robotic music system with plug-and-play ease-of-use, a highly modular design, and high-performance components. Our ultimate goal for the NotomotoN was to create a teaching and performance tool that can serve both as an introduction to musical robotics for those unfamiliar with the field and as a powerful instrument that pushes the boundaries of robotic music performance.

The NotomotoN's ease of use lies in part in its use of the new Arduino-based USB MIDI handler. Users can simply plug into the robot's USB jack and select the robot as a standard MIDI device. Unlike many previous robotic music systems, highly specialized custom software is not needed. Section 2 focuses on the MIDI handling as well as custom-designed high-performance hardware utilized in the NotomotoN. Section 3 presents a detailed performance analysis of the solenoid striker assemblies utilized on the NotomotoN, with special focus dedicated to striker latency, force, and rate of action. Finally, section 4 focuses on live-performance and educational uses of the NotomotoN.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

1.1 Robotic Percussion: Related Work

Automated percussion systems can be traced back to mechanically sequenced instruments related to player pianos. However, Trimpin is one of the pioneers of computerized robotic percussion instruments and since the 1970's has created a diverse array of percussion works [8]. Another significant early innovator in the field is Godfried-Willem Raes of the Logos Foundation: his works from the early 1980's, though, were "soundsculptures in the full sense: not real musical instruments, and not playable[1]".

The 1990's and 2000's saw an explosion in robotic percussion research with a focus on interactivity: Eric Singer's LEMUR creations were truly live-performance capable [7], and both the first author [4] and Gil Weinberg [10] explored performer/robot interaction. In recent years, such performances as Trimpin's 2007 collaboration with the Kronos Quartet and the KarmetiK Machine Orchestra [2] have highlighted the benefits of fully integrated robotic music systems which do away with the need for in-depth calibration and focus on ease of configuration. The success of Pat Matheny's Orchestrion[2] with Eric Singer's instruments has proven that the public is ready for robots on stage. The KarmetiK NotomotoN builds on the previous work done in the field of musical robotics to further allow for faster deployment time and ease of use.

2. TECHNICAL DETAILS

The KarmetiK NotomotoN is a robotic drum featuring twin drum heads, a metal body, and 18 solenoid beater assemblies (see Section 2.2). Each NotomotoN is a fully integrated unit: all power supply components and electronics are self-contained within the drum's body, and all beater assemblies are mounted to the robot's dual circular superstructures. The NotomotoN is intended to be usable as a desktop unit.

Figure 1. The KarmetiK NotomotoN

[1] http://www.logosfoundation.org/g_texts/ibart-leonardo.html
[2] http://www.patmetheny.com/orchestrioninfo/

2.1 Hardware

The NotomotoN has two types of beater assemblies each utilizing a different actuation technique (see Section 3.1). The two types of beaters used were called the TrimpTron and the KalTron. The solenoid beater assemblies utilized CNC manufacturing techniques to allow for interchangeability, precision, and consistency. In addition to the listed solenoid drumbeater assemblies, extra electronics and wiring have been allocated to allow for the attachment of additional actuators.

2.1.1 The TrimpTron

The TrimpTron makes use of a rotary solenoid mounted perpendicular to the NotomotoN's drumheads. Its aluminum mounting bracket can be rotated on the robot's superstructure, allowing for a wide variety of timbres as the drum head is struck in different places.

Figure 2. TrimpTron Solenoids

2.1.2 The KalTron

The KalTron drumbeater (See Figure 3) assembly uses a modified pull-type solenoid arranged such that its linear motion is converted to rotational motion. The pull solenoids used in the KalTrons allow for very rapid-fire actuation: high-speed rolls are possible using the KalTrons.

Figure 3. KalTrons on the NotomotoN's Superstructure

2.1.3 Additional Actuators

In addition to the TrimpTrons, and the KalTron, the NotomotoN has electronics and wiring for up to six additional actuators. The use of additional actuators can prove greatly useful in a teaching environment, as discussed in Section 5.2.

2.2 Power and Electronics

A chief design objective in the conception of the NotomotoN was the integration of a 12V DC power supply and electronics in the main body of the drum. Prior experience indicated that, in a teaching and performance environment, having numerous external enclosures for power and communication resulted in a highly entropic cabling situation. The NotomotoN, on the other hand, needs only two cables: one for power and a USB cable for communication. This highly flexible configuration allows for those lacking extensive experience with musical robotic systems to simply plug in and begin composing and performing.

2.2.1 MIDI Subsystem

The AVR-based Arduino physical computing platform was chosen as the means of communication between users' computers and the NotomotoN's actuators. The Arduino was chosen due to its open-source nature, extensive documentation, and compact size. A custom-developed daughterboard was utilized to convert 5V control voltages from the AVR to 24V actuator signals. The daughterboard (dubbed the KarmetiK Trinity board) connects to the Arduino as a shield, sitting directly on the header pins of the Arduino board. The Arduino and Trinity board combination proved quite compact, allowing the entire assembly to rest next to the power supply, completely concealed inside the NotomotoN's drum body. To communicate with the Arduino board, users plug in to a USB jack on the NotomotoN's outer shell.

2.2.2 Power Subsystem

In order to allow for the 24V DC power supply to fit within the NotomotoN's body, an aluminum cradle assembly was built. The cradle seats the power supply and is secured to the drum body's sides. As with the communication assembly, all power components are concealed within the NotomotoN's metal body. To power the actuators on the NotomotoN, users plug in to the female IEC connector.

2.3 Software

In an effort to enable plug-and-play ease-of-use on the NotomotoN, MIDI was chosen as the main communication protocol between the robot and performers' computers. Thanks to the Arduino's adoption of the user-configurable ATMega8U2 USB to serial converter, MIDI over USB HID proved possible. For performance and educational use, custom-build middleware is used. This middleware allows users and administrators to customize outgoing messages, preventing potentially harmful messages from reaching the NotomotoN and allowing messages to be analyzed and rerouted as desired. Additionally, this middleware serves as an OSC to midi translator, opening up the possibilities of communicating with the NotomotoN from many other software and hardware platforms.

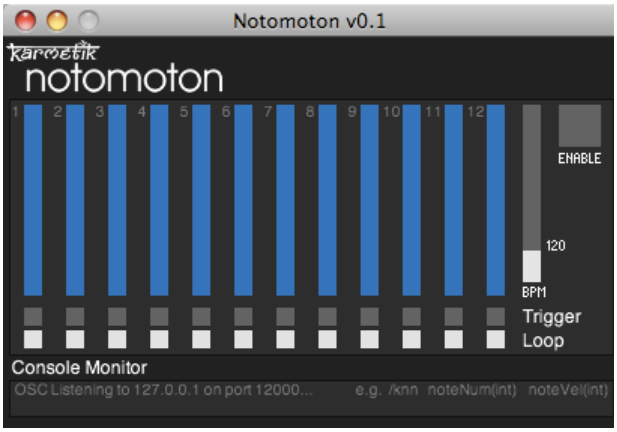


Figure 4. KarmetiK NotomotoN Software GUI

3. PERFORMANCE ANALYSIS

In order to optimize the actuators' behavior, experiments testing the drumbeaters' speed, dynamics, and latency were conducted. These experiments allowed us to understand the advantages and disadvantages of both beater types (see Section 3.4) and optimize their use in an educational and performance context. All tests were conducted with the beaters striking 20mm away from a piezo sensor on a 20cm diameter drum head. Tests involving the TrimpTron were performed with the drum beater 15mm above the drum head. Due to design differences, KalTron tests were performed with the drum beater 10mm above the drum head.

3.1 Actuator Speed Test

To test the frequency at which the actuator assemblies were capable of striking surfaces, we conducted actuator speed tests. Incrementally decreasing intervals between MIDI Note On events were sent to the NotomotoN, with the fastest strike frequency recorded. Table 1 and Figure 5 illustrate the data.

Table 1. TrimpTron and KalTron Actuator Strike Frequency

MIDI Velocity	TrimpTron Strike Frequency	4. KalTron Strike Frequency
127	18.69Hz	20.40Hz
64	30.21Hz	43.10Hz
50	39.9Hz	68.90Hz

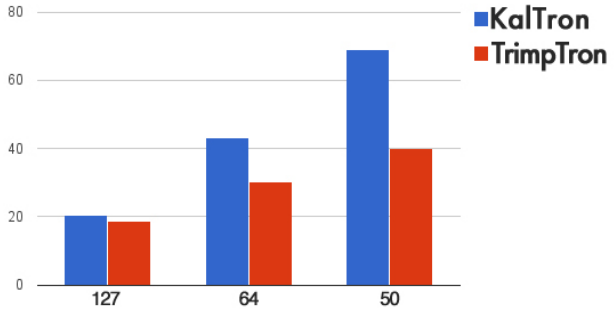


Figure 5. Actuator Speed Tests (X-axis = MIDI velocity; Y-axis = Strikes/Second)

3.2 Actuator Dynamics Test

Figure 6 illustrates output velocity plotted against the actual velocity of the actuators. The purpose of this experiment was to determine the response curve of the actuators as output velocities increased in order to better understand the real-world behavior of the actuators. MIDI notes were sent 1 second apart to allow for the system to return to equilibrium before subsequent note on events were sent.

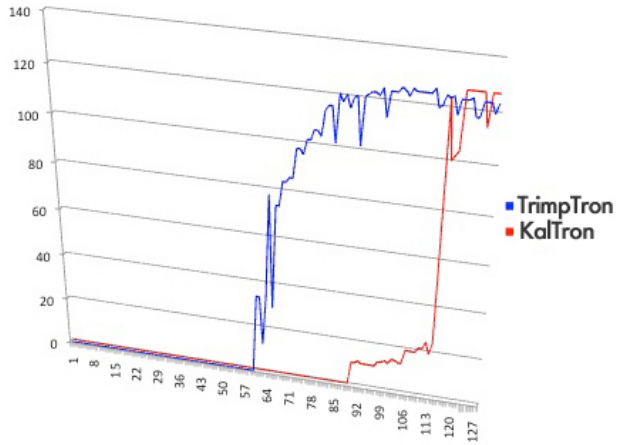


Figure 6. Actuator Velocity vs. Measured Output Velocity (X-axis = output velocity; Y-axis = Actuator velocity)

3.3 Actuator Delay Test

Actuator latency is defined here as the time between a MIDI message being sent from a server to the time at which the actuator makes contact with the surface to be struck. Table 1 shows the results under testing conditions (20mm above the drum head on the TrimpTron and 10mm above the drum head on the KalTron).

Table 2. Actuator Delay Time With Differing Velocities

MIDI Velocity	TrimpTron Latency	KalTron Latency
127	0.041 seconds	0.034 seconds
64	0.052 seconds	0.032 seconds
50	0.058 seconds	0.029 seconds

4.1 Test Conclusions

The Trimpin Rotary Solenoid Assembly was found to compliment the KalTron: while the KalTron is capable of very rapid strike frequencies, its dynamic response curve is less consistent than the TrimpTron. The TrimpTron, then, can serve as a more dynamically expressive actuator while the KalTron plays the role of performing high speed drum rolls.

5. PERFORMANCE AND EDUCATION

5.1 Live Performance Use

Through testing and use in a live musical context, the NotomotoN has been found to excel as a performance tool thanks to its ability to respond rapidly and dynamically to user commands (see Section 3) as well as its compact size and fast setup time. In a live performance context, the NotomotoN has been tested and played with a diverse array of software and hardware-based musical interfaces, including the RadioDrum [5], the Kinetic Engine [1], the ESitar [4], the Helio [6], and the

Arduinome [9]. Thanks to its ability to respond subtly to parametric control, the NotomotoN was found to be highly expressive when used by the greatly differing above performance interfaces.

5.2 Educational Applications

The NotomotoN was designed to serve as the centerpiece for a musical robotic education curriculum [3]. The unit works both as a completely integrated robotic instrument with multiple beater types and as a central module onto which six additional drum-beater mechanisms can be attached. In its role as a central module with external actuators, the NotomotoN serves as a means by which students can rapidly test new beater designs and test existing actuators against new materials such as found objects and drums. In an educational context, the NotomotoN has been used to introduce students to musical robotics and allow them to begin composing for them rapidly.

6. CONCLUSION

The KarmetiK NotomotoN is a new breed of musical robot allowing for rapid deployment and heretofore unseen ease of use in a self-contained package. For performance scenarios, the NotomotoN's consistent and high-performance actuators allow for highly expressive musical performance capable of taking advantage of the expressivity offered by new musical performance interfaces. While the NotomotoN excels in performance situations, it also serves as an excellent educational package due to its ease of use and extensible actuator options. To foster further work in the field of musical robotics, the KarmetiK NotomotoN has been made available to all performers and educational institutions. Please visit http://www.notomoton.com for further information about the KarmetiK NotomotoN.

7. REFERENCES

[1] Eigenfeldt, A. *The Evolution of Evolutionary Software: Intelligent Rhythm Generation in Kinetic Engine.* In *Proceedings of the EvoWorkshows 2009 on Applications of Evolutionary Computing*. 2009. Berlin.

[2] Kapur, A., et al. *The Machine Orchestra*. In *Proceedings of the International Computer Music Conference*. 2010. New York City.

[3] Kapur, A., Darling, M. *A Pedagogical Paradigm for Musical Robotics*. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 2010. Sydney.

[4] Kapur, A., *Digitizing North Indian Music: Preservation and Extension using Multimodal Sensor Systems, Machine Learning and Robotics*. VDM Verlag Dr. Muller, Germany, 2008.

[5] Mathews, M., and Schloss, W. Andrew. *The Radio Drum as a Synthesizer Controller*. In *Proceedings of the International Computer Music Conference.* 1989. Ohio State University.

[6] Murphy, J., Kapur, A., Burgin, C. *The Helio: A Study of Membrane Potentiometers and Long Force Sensing Resistors for Musical Interfaces.* In *Proceedings of the International Conference on New Interfaces for Musical Expression*. 2010. Sydney.

[7] Singer, E., et al. *LEMUR's Musical Robots*. In *International Conference on New Interfaces for Musical Expression*. 2004. Hamamatsu, Japan.

[8] Trimpin, *SoundSculptures: Five Examples*. 2000, Munich MGM MediaGruppe Munchen.

[9] Vallis, O., Hochenbaum, J., and Kapur, A., *A Shift Towards Iterative and Open-Source Design for Musical Interfaces.* In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Sydney, Australia. 2010.

[10] Weinberg, G., S. Driscoll, and T. Thatcher. *Jam'aa - A Middle Eastern Percussion Ensemble for Human and Robotic Players*. In *International Computer Music Conference*. 2006. New Orleans.

The Manipuller: Strings Manipulation and Multi-Dimensional Force Sensing

Adrián Barenca
Digital Media and Arts Research Centre
Dept. of Computer Science and Information Systems
Faculty of Science and Engineering
University of Limerick
adrianbarenca@gmail.com

Giuseppe Torre
Digital Media and Arts Research Centre
Dept. of Computer Science and Information Systems
Faculty of Science and Engineering
University of Limerick
giuseppe.torre@ul.ie

ABSTRACT
The Manipuller is a novel Gestural Controller based on strings manipulation and multi-dimensional force sensing technology. This paper describes its motivation, design and operational principles along with some of its musical applications. Finally the results of a preliminary usability test are presented and discussed.

Keywords
Gestural Controller, Strings, Manipulation, Force Sensing.

1. STRINGS AND FORCE SENSING
The integration of strings and force sensors within Gestural Controllers is a powerful combination towards higher grades of feedback and expressivity in a live performance. Strings provide excellent tactile feedback, which is crucial for the performer and also a key factor for the audience to follow the performance. Furthermore the manipulation of a string requires some physical effort, which helps to the funneling of performance expressivity. Hence strings manipulation has intrinsic expressive qualities, for both performer and audience; somehow reflecting the musical duality of tension-release. Strings are common items of our daily life, and have been since ancient times. That makes them very familiar to everybody, turning their manipulation into a very intuitive action: grabbing, pulling, twisting, plucking, etc. Therefore the learning and adaptation time in a string based Gestural Controller should become minimal for the performer.

In the other hand, the use of force sensors within Gestural Controllers aims to register effectively the dynamic physical effort of the performance and transmit its expressivity to an audience. In fact, since effort corresponds to power, and power is defined as the product of the force and the speed (energy over time), then force sensors should effectively register this effort during live performance.

1.1. Related Gestural Controllers
The Web [7] uses interconnected string segments within a hexagonal frame, each one equipped with a tension sensor. The performer pulls a segment, which influences its neighbour segments, distributing the tension over the whole web. The registered tension of each segment represents an input variable to the sound synthesis system. The tension variations were then mapped to produce sound of complex changes in timbre.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

The way *The Web* worked proved to be very easy and effective: grab the strings, pull the segments, and hear immediate complex changes in timbre. Within a short time, musicians with no theoretical knowledge about the system were able to master a great amount of control over the timbres. The strings provide good tactile feedback and the tension force sensors are highly responsive to the strings manipulation.

Some potential limitations of *The Web* are: small manipulation range, which may result in no appreciable visual feedback for the audience; complex synthesis with simple gestures, affecting to the grade of control over the output sound; the physical distribution and the interconnection of the strings, which makes the system to work as a whole where segments cannot be isolated completely.

The Soundnet [9] is a large scale version of *The Web*, allowing several performers to climb all over its stringed structure. Eleven transducers are distributed along the 11x11meters structure to detect stretching and movement. The data extracted from these is then mapped to control filter parameters and granular synthesis. Whilst *The Soundnet* provides good feedback to the audience, the control capabilities in a performance are subject to extreme physical effort. Furthermore the requirement of a large space makes it very hard for performers to rehearse and master *The Soundnet*.

The concept of the *Pullka* [5] is based on a single fixed string with one strain gauge which measures the tension as the performer pulls the string behind one of the two bridges. In fact it was thought as a two person interface, each actuating over one string end. This simple controller would reduce the gestures sensing to only one parameter (the string tension), representing a real challenge towards sound mapping. Nevertheless there seems to be no further documentation or tangible evidence claiming the *Pullka* was ever further developed or implemented.

The *Strimidilator* [3] senses the deviation and the vibration of a set of four parallel strings manipulated by pulling and plucking actions. The vibration is sensed in two strings using electric guitar pickup coils, and the deviation is sensed in the other two by attaching a variable resistor to one of the fixed ends. All parameters are converted to MIDI messages. However the main instrument controls correspond to buttons and knobs placed in a box adjacent to the frame. The sensed string parameters are mapped to envelope or dynamics, while the buttons and knobs control the main parameters such as note-on, note off, and variation mode. This means that the four strings will be mainly played or manipulated with just one hand, since the other one will have to spend most of the time on the knobs and buttons. Direct tactile force feedback is provided only on the two strings whose deviation is being sensed. However the attachment mechanism of the deviation transducer to the string seems very poor and unreliable, causing significant damping and affecting the vibration mode of the string.

1.2. Towards Multi-Dimensional Sensing

In the Gestural Controllers described above, the force is only sensed in one dimension, limiting the kinetic range of tracked gestures. The combination of strings and force sensing has to be fully exploited in terms of the range of gesture parameters to be tracked for sound mapping. The path to improve the gestures tracking range of such controllers passes through adding the capability to sense the forces in more than one dimension. Hence the combination of strings, force and multidimensional sensing for musical applications is the *leitmotiv* of the developed prototype of Gestural Controller: The Manipuller.

2. SYSTEM OVERVIEW

Figure 1 shows the general block diagram of the Manipuller.

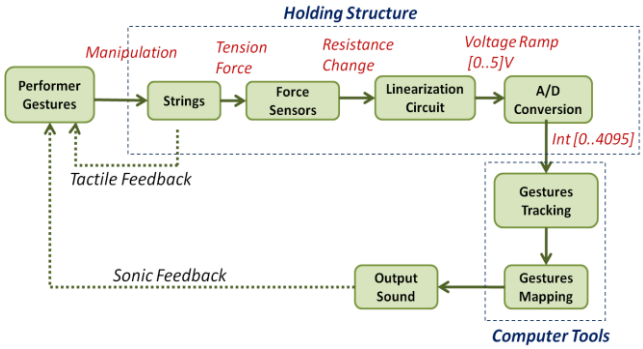


Figure 1. General Block Diagram

The primary interface consists of four parallel elastic strings manipulated by the performer. The holding structure is a semi-open wooden frame containing the strings, sensors and electronic circuitry (Figure 2).

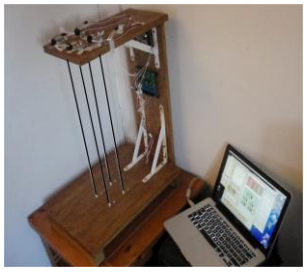


Figure 2. The Manipuller System

The performer actuates over the four strings by manipulating them in group or independently. Each string tension is independently sensed at both top and bottom ends by a pair of Force Sensing Resistors (FSRs) [6]. The force sensors are placed on the opposite sides of the top and bottom platforms. The string is vertically guided and clamped at the sensors sides to a mechanical actuator which will transmit the pulling force *proportionally* over the sensing area. Each sensor's output is linearized and converted to voltage by hardware means [6] before being digitized.

The Analogue to Digital conversion is carried out by programmable microcontroller board Arduino Mega [1]. Its USB connection to a laptop computer also provides the power supply for the linearization board. The eight analogue inputs are oversampled and decimated [2] to produce integer values between 0 and 4095 (virtual 12bit), which are then scaled to floats ranging from 0 to 1. Valid data from each sensor is ready for computer processing every 10ms.

3. GESTURES TRACKING & MAPPING

Gestures coding is achieved by the algorithms and numerical interpretation of the different gestures, such as 3D position tracking, pull-release, angular speed and sign of turn. All coding is carried out within the digital domain by computer means. In this case the software tool chosen was the visual programming environment MaxMSP [4].

3.1. Simple Parameters

The first step to test the system is to focus on single string parameters. For instance, if we take String 1 and use the top (S_{1T}) and bottom (S_{1B}) sensor readings we can map the differential (S_{1T} - S_{1B}) to the frequency of a pure tone. Therefore pulling the string downwards results in more tension being registered at the top sensor, resulting in higher pitches. Similarly, pulling the string upwards results in lower pitches.

As an example of more complex mapping, we carried out a Frequency Modulation synthesis, where the carrier frequency is given by S_{1T}, the modulation index by (S_{1T} - S_{1B})*10, and the harmonicity ratio by S_{1T}/S_{1B}. To obtain meaningful sound S_{1T} and S_{1B} were rescaled from [0, 1] to [100, 3500] Hz. Similar type of combinations and procedures were used for other synthesis methods, such as Amplitude and Ring Modulation.

Furthermore it is possible to assign different sound or control parameters to each string. One of the recently developed applications is to use the strings to control different aspects of a sample-buffer-based granular synthesis: String 1 would determine the rate at which grains are triggered; String 2 would affect the grains pitch; and String 3 and 4 would determine grain size and sample buffer offset respectively.

3.2. Pull-Release

It is possible to detect whether the string is being pulled or released by means of gradient calculations. An effective application is a *sound file playback control*, where pulling the string plays the sound forward, and releasing it does it backwards. The average value registered by the force sensors control the ratio of the playback speed. The approach is as follows: if F_0 is the registered pulling value at a given time t_0, and F_1 is that value at a later time $t_1 = t_0 + \Delta t$, then: a) if $F_1 > F_0$, then the string is being *pulled* and the playback is *forward*; b) if $F_1 < F_0$, then the string is being *released* and the playback is *backwards*; *c)* if $F_1 = F_0 \neq 0$, then the string is being *held* and the playback *continues* at the speed given by the sensors; and *d)* if $F_1 = F_0 = 0$, then the string is in *standby* (no manipulation) and the playback is *stopped*.

On hold, the playback direction corresponds to the previous state. For instance, if the string is being released and then set on hold, the playback would continue backwards. In a practical implementation it will be necessary to include determined threshold values and marginal limits in the conditions.

Similarly the differential values of each string can be mapped to a portion of a *pitches scale*. The first string is mapped to the lower pitches; the second to the low-mid pitches; the third to the mid-high; and the fourth to the higher pitches. Therefore, when a string is pulled and then held, a pitch of the assigned scale portion is played accordingly to the top-bottom differential value. The velocity of the pitch is determined by the average string tension. Hence manipulating all strings results in chords of separated pitches.

3.3. Multiple Strings

The most interesting feature of the Manipuller lies in the configuration of the strings within the frame. When all strings are grabbed and manipulated at once, the combinational reading of the differential tension forces of each string can be used to track spatial gestures, such as position vector, direction and sign of the force, or the angular speed for a circular gesture. These gestures, when effectively mapped into musical

parameters, provide excellent correlation between performance and output sound.

In order to obtain a spatial position vector which would correlate with the kinetics of the strings manipulation, the Manipuller has to be placed within the Three-Dimensional Space by defining a Cartesian Coordinates Reference System (Figure 4).

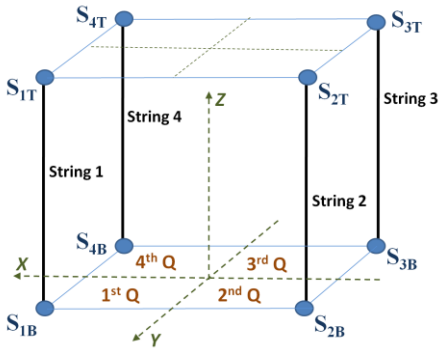


Figure 3. Configuration within the Reference System

Since each string will correspond to one quadrant, we can assign one base vector from the X-Y plane per string. Each base vector will be scaled accordingly to the average tension (AVG_n) registered by the top and bottom sensors (S_{nT}, S_{nB}). The virtual Z coordinate will be given by the differential top-bottom tensions at each string ($d_n = S_{nT} - S_{nB}$). Since the sensor values are float numbers between 0 and 1, then the range for the average values is [0, 1], and [-1, 1] for the differential.

Table 1. Associated Base Vector and Spatial Coordinates

String	(X, Y) Vector	(X, Y, Z) Coordinates
1st	(1, 1)	R_1=(AVG_1, AVG_1, d_1)
2nd	(-1, 1)	R_2= (-AVG_2, AVG_2, d_2)
3rd	(-1, -1)	R_3= (-AVG_3, -AVG_3, d_3)
4th	(1, -1)	R_4= (AVG_4, -AVG_4, d_4)

For instance if String 1 is pulled, this results in a 45 degrees vector in the 1st Quadrant, whose (x, y) coordinates are (AVG_1, AVG_1). Similarly, pulling String 2 results in a 45 degrees vector in the 2nd Quadrant, with coordinates (-AVG_2, AVG_2). Hence when Strings 1 and 2 are pulled simultaneously the result is a vector which is the algebraic sum of their respective Cartesian coordinates. Hence by grabbing and pulling all four strings at once it is possible to track the kinetics of the manipulation (Figure 5).

Since it is possible to know the X-Y coordinates, then the evolution of the position vector over time can provide information about whether a circular movement is taking place, its angular speed, and its sign of turn. For instance, when a circular gesture is detected, a sound file can be played back or forward accordingly to the sign and value of the angular speed (rpm) detected by the algorithm. This application would emulate a *turntable*. It is also possible to incorporate a lap counter to control reverberation or delay times.

Furthermore the capability of spatial tracking makes the system an ideal candidate as a live performance tool for sound spatialisation control in a 3D speaker setup, such as Ambisonics [8]. Figure 6 shows a set up of eight speakers equidistant in the horizontal plane but with different elevations. In this case, the resultant vector places the source within the 2nd Quadrant of the Ambisonics monitor; the resultant *Z* indicates there is a significant elevation. This correlates with the output levels shown in the indicator panel.

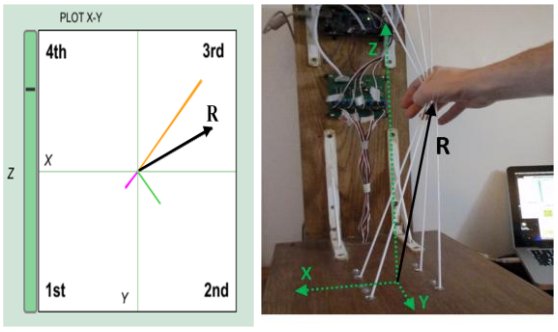


Figure 5. Four Strings Spatial Tracking

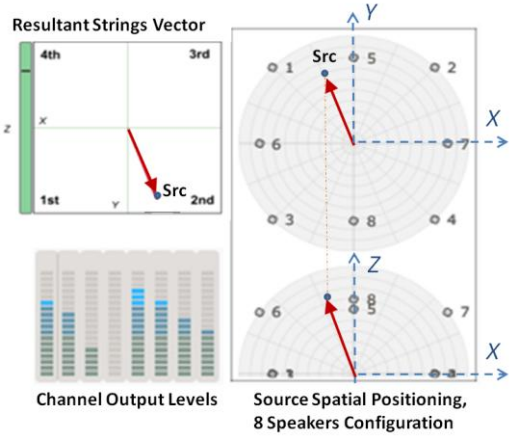


Figure 6. Sound Source 3D Spatialization

4. EVALUATION

The evaluation of the Manipuller should provide valuable information about crucial aspects such as *Learnability*, *Explorability*, *Feature Controllability*, *and Timing Controllability* [10].

In general terms we aim to estimate the amount of time needed to learn how to control a performance with the Manipuller; to evaluate its intrinsic feedback properties, the degrees of freedom and the number of different gestures that can be tracked; to determine its potential in terms of performance expressivity; to test its accuracy, resolution and manipulation range; and to hear suggestions about new musical applications which could be added to the mapping strategy.

4.1. Method

A usability test was designed to evaluate the main features of the Manipuller. Participants will compare the stringed interface against a Game Pad standard controller by performing determined musical tasks.

The Game Pad emulates the (X, Y, Z) coordinates by actuating on its two joystick controllers (X-Y for the left, Z for the right controller). However, due to its intrinsic characteristics, the Game Pad cannot effectively emulate all the musical applications implemented for the Manipuller. For this reason, and to keep the duration of each test under 30 minutes, only three musical tasks were chosen for the test (section 4.3).

4.2. Set Up

A total of seven individuals with diverse musical and technological background kindly agreed to participate in the usability test without any economic compensation. Their ages ranged from 23 to 39 years (mean=29). Each individual test took around 25 minutes, and consisted on a brief introduction to

the setup and to the operation principles of the Manipuller. The participant was then asked to perform three tasks alternating the Game Pad and the Manipuller. At the end of the session the participant was asked to complete a comparative questionnaire about relevant aspects such as expressivity, feedback, learnability, responsiveness and control accuracy. An additional page was provided for additional comments.

4.3. Musical Tasks

The first task allowed the user to freely perform on both controllers. The task consisted on a Frequency Modulation patch where (X, Y, Z) were mapped to carrier frequency, modulation index and harmonicity ratio.

The second task used a patch to map the coordinates to a parametric filter (cut off, Q, Gain). The input chosen for the filter was a white noise source. The user was asked to freely manipulate the controllers (first the Game Pad) with special attention to the correlation of the gestures with the produced sound.

In the third task the participant was asked to emulate a turntable by performing circular gestures on the Game Pad left controller and then in the Manipuller by grabbing all four strings and making sure the circular gesture was well defined. A pair of sound files would then play forward or backwards accordingly to the angular speed and the sign of turn of the circular gesture.

4.4. Results

At the end of each session the participant was asked to fill in a questionnaire, which consisted on a header section containing information about the age, musical instruments and preferences; a main section with eighteen affirmative sentences relevant to the tests which had to be rated in a scale from 1 to 5 (1: Strongly Disagree, 2: Disagree, 3: Neutral, 4: Agree, 5: Strongly Agree); and a closing section with two questions about possible additional gestures and other musical applications, and a space for any additional comments. The mean results from the main questionnaire were displayed in a bar graph for later analysis (Figure 8).

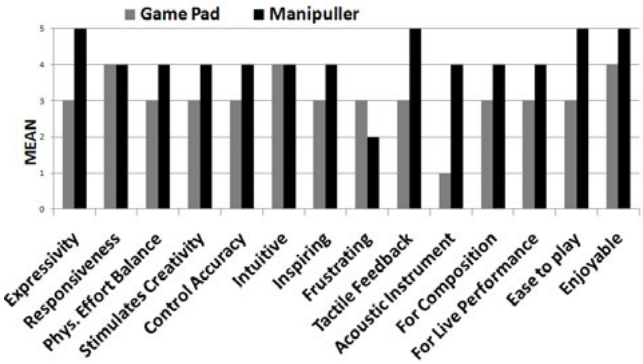


Figure 8. Usability Test Results

5. DISCUSSION

Overall the Manipuller received more positive results than the Game Pad, particularly in terms of expressivity, tactile feedback, "feel-like" acoustic instrument, and ease to play. With less difference, but still better rated, the Manipuller provided better balance between the delivered physical effort and the generated sound, and better perceived control accuracy over the output. Similarly, it also rated higher as a stimulus for creativity, and resulted more inspiring (and less frustrating) than the Game Pad. Participants found the Manipuller more enjoyable, and also preferred it for music composition, live performance or other artistic purposes. Both controllers got the same positive assessment in responsiveness and intuitiveness. Nevertheless we can then say the Manipuller is as responsive and intuitive to use as a Game Pad.

The comments left by the participants provided valuable suggestions to improve the Manipuller with additional gestures detection and other musical applications. There were suggestions about the possibility of actuating over the strings by plucking; to incorporate motion detection to reinforce the tracking of hand movement and position; or to explore the possibility of detecting the hand position over the string while being continuously pulled or held. Some participants suggested some other musical applications, such as the addition of an analogical input for an external line input to use the Manipuller as a real time sound modifier. One participant reported the Manipuller as confusing in the way it handled more than one parameter for musical mapping. Another participant commented that the physical effort needed to actuate over the Manipuller would eventually lean him towards the Game Pad.

6. CONCLUSION

We have presented the Manipuller, a novel gestural controller based on strings manipulation and force sensing technology. It combines the differential tension force sensing of fixed strings to track Three-Dimensional spatial gestures. The combination of strings and force sensing adds tactile feedback and physical effort, which are crucial to achieve high grades of expressivity during live performance. The Manipuller is highly responsive, flexible, intuitive and relatively easy to use. It provides a meaningful correlation between the performer's actions and the perceived output sound.

7. ACKNOWLEDGEMENTS

We would like to thank NAIRTL funding body for the research opportunity given. Special thanks to Adrian Freed for his valuable feedback towards the camera-ready paper.

8. REFERENCES

[1] Arduino. Arduino Mega Board, 2007. Retrieved 1st Aug. 2010: http://arduino.cc/en/Main/ArduinoBoardMega.

[2] Atmel. AVR121: Enhancing ADC Resolution by Oversampling. Application Note Rev. 8003A-AVR-09/05

[3] Baalman, M. A. J. The STRIMIDILATOR: a String Controlled MIDI-Instrument. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, (Montreal, Canada, 2003).

[4] Cycling'74. MaxMSP Graphic Programming Environment, 2009. Retrieved 10th August 2010: http://www.cycling74.com.

[5] Fels, S. and Vogt, F. Tooka: Explorations of Two Person Instruments. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression,* May 24-26, (Dublin, Ireland, 2002).

[6] Flexiforce. Flexiforce Sensors User Manual, 2001. Retrieved 16th April 2010: http://www.tekscan.com/pdfs/FlexiforceUserManual.pdf.

[7] Krefeld, V. The Hand in the Web: An Interview with Michel Waisvisz. *Computer Music Journal*, 14 (2), 1990.

[8] Malham, D. G., and Myatt, A. 3-D Sound Spatialization Using Ambisonic Techniques. *Computer Music Journal*, 19(4), 1995.

[9] Tanaka, A. Musical Performance Practice on Sensor-based Instruments. Faculty of Media Arts and Sciences, Chukyo University, Toyota-Shi, Japan, 2000.

[10] Wanderley, M. M., and Orio, N. Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal*, Vol. 26, No. 3, New Performance Interfaces (Autumn), pp. 62-76 Published by The MIT Press, 2002.

Mapping Objects with the Surface Editor

Alain Crevoisier
Haute Ecole de Musique de Genève (HEM)
Rue de l'Arquebuse 12, CP 5155
CH-1211 GENEVE 11
alain.crevoisier@hesge.ch

Cécile Picard-Limpens
Haute Ecole de Musique de Genève (HEM)
Rue de l'Arquebuse 12, CP 5155
CH-1211 GENEVE 11
ccl.picard@gmail.com

ABSTRACT
The Surface Editor is a software tool for creating control interfaces and mapping input actions to OSC or MIDI actions very easily and intuitively. Originally conceived to be used with a tactile interface, the Surface Editor has been extended to support the creation of graspable interfaces as well. This paper presents a new framework for the generic mapping of user actions with graspable objects on a surface. We also present a system for detecting touch on thin objects, allowing for extended interactive possibilities. The Surface Editor is not limited to a particular tracking system though, and the generic mapping approach for objects can have a broader use with various input interfaces supporting touch and/or objects.

Keywords
NIME, mapping, interaction, user-defined interfaces, tangibles, graspable interfaces.

1. INTRODUCTION
Projects on tangible interfaces [11] have grown since the last decade, and a large part of them concerns new ways of performing music [4]. More recently, the concept of Natural User Interface (NUI) has been used and refers to "*a user interface that is effectively invisible*" [7]. Our work gathers achievements in the field of graspable interfaces, NUI technologies and new interfaces for musical expressions. We extended and adapted our previous research on tactile interfaces for the use of graspable objects, leading to a concept of interaction based on combining the manipulation of objects with touch sensing. Detecting fingers touching the surface of an object offers the opportunity to intuitively trigger actions by simply tapping on the object. For our experiments, we have used a multitouch technology that makes possible to transform any flat surface into a multitouch device [1] and ReacTIVision, a well known Computer Vision tool for identifying objects and tracking their position and orientation using visual markers [3]. However, the framework we have developed for mapping objects is not relying on a particular technology and any input interface supporting the TUIO protocol [5] can be used, although the touch-on-object information may not be available in all cases.

Complex mapping structures can be determined with the Surface Editor thanks to the possibility to assign several actions for an object and also to set rules for the conditional activation of an action or a group of actions [6]. This is particularly useful for exploring new mapping strategies between input gestures and musical actions. The finality of our study is to propose a generic mapping tool for setting up and configuring graspable interfaces adaptable for any use case scenarios.

2. RELATED WORK
A new trend in the field of Human Computer Interaction (HCI) has been observed this last decade: interfaces are more and more adapted to our ways of experiencing the world. As an example, multitouch interaction is gradually supplanting the use of the traditional computer mouse as control component. Another example is given by graspable interfaces. As noticed in 1995 by Fitzmaurice et al. [2], a graspable interface exploits not only our well-developed, everyday haptic-tactile skills for physical object manipulation, but also our sharp spatial reasoning capacities. In addition, it enables multi-person, collaborative use. Playing with objects, combining them in order to create something new, is a way of showing our interpretation of the world [13]. Further, the use of objects as controllers give a persistent representation of what is manipulated [8]. The idea is not only to handle objects independently from each other but also putting them into relation with one another. As outlined by Wanderley et al. [12], performing computer music with controllers is closely related to the notion of mapping. Indeed, analyzing the influence of mapping on the performance of digital musical instruments or systematically defining mappings to relate controller variables to synthesis inputs remain crucial.

Among the many tangible interfaces projects that have been developed over theses two decades [4], the Reactable [3] is by far the most remarkable, and the closest to our study. The Reactable takes its origin from studies on musical performance and the interest in developing interfaces for the real-time creation and exploration of music. It uses visual marker recognition (VMR) for object identification. The specific set-up of the ReacTable includes a semi-transparent surface with a camera and projector behind it.

Our method differs in many ways. First of all, it uses ordinary tables and surfaces for interaction. Custom made tables can be visually attractive, but they have the main drawback of not being suitable for widely use. Second, contrary to the majority of similar projects that require an image to be projected on the table, it is only an option in our case. This makes the system easily transportable everywhere. Already with one of the first projects on tangible interfaces, Audiopad [9], cumbersome video projection was seen as a technical limitation. Third, none of similar projects allows for detecting fingers touching on an object. In that way, our system gives the opportunity to explore new interactive possibilities. Finally, as pointed by the SenseTable project [8], interacting with a large amount of information with a finite number of physical objects remains challenging. For this purpose, we propose to leave to the user the ability to define and adjust the actions in response to his interaction, so that he can design the mapping that fits his need.

3. OBJECT DEFINITION

An object is defined as any element that can be grasped and put on a surface. Objects must be identified somehow. Most systems require a tag to be attached to the objects in order to facilitate the object recognition and identification. Objects are defined by their attributes: Object ID (Tag ID), Size, Tag position, and Type. The last attribute is useful to set families of objects. It is up to the users to give a meaning and function to the objects. For instance, one could attribute characteristics related to sound and music to objects, such as sound sources, sound modifiers (effects), loop players, volume button, track selector, or any kind of processing parameter.

4. OBJECT RELATIONS AND DERIVED PARAMETERS

4.1 No Relation

In this case, the object do not relate with any other object, even if there might be some others in the neighborhood. The only parameters are the absolute position and rotation angle of the object. These may be considered as the intrinsic parameters of an object, since they exist in all cases, independently of its relation with other objects.

Figure 1. Neighborhood relation (left) and Object-in-Object relation (right).

4.2 Neighborhood

This is the simplest relation between two objects (Figure 1, left). Beside the intrinsic parameters peculiar to each object, new parameters are derived from the relative position of the two objects:

- Delta X
- Delta Y
- Distance
- Angle

Except for the angle between the two objects, which is always calculated from center-to-center, the other parameters are calculated both from center-to-center and from edge-to-edge, providing a total of seven derived parameter.

4.3 Object in Object

This relation exists for instance if a thin object is sufficiently big in size to contain one or more smaller objects (Figure 1, right). In this case the derived parameters are given by the relative position of the smaller object inside the bigger one.

4.4 Selective Relations

In most cases, we don't want an object to relate with all others. For instance, we may want that an object can relate only with a particular kind of objects, or only within a certain distance. In order to consider selective relations, we need to define conditions under which two or more objects can enter in relation. For this purpose, we introduce the notion of *filters*. Filters can be applied either to the Object ID, the Object Type, the intrinsic parameters, or the derived parameters. In addition, several filters can be combined to define more selective conditions.

5. CHAINS

Chains are created when two or more objects enter in relation. They can also be seen as sequences of objects ordered in the 2D space. Figure 2 shows two examples of chains where objects are close to each other. However, objects do not need to be close to form a chain and the two configurations in Figure 1 are also valid examples of chains. A chain exists as long as the objects are in relation and that the conditions set for the filters are satisfied. A chain is defining a new entity, which extends the object's characteristics. As a consequence, objects belonging to a chain get new attributes:

- The ID of the chain they belong to (Sequence ID)
- Their position in the chain (Ordering number)
- The number of elements in the chain (Chain length)

For simplicity, we did not individuated branches in a chain. For this reason, several objects may get the same ordering number (see section 5.2 for details on attribution).

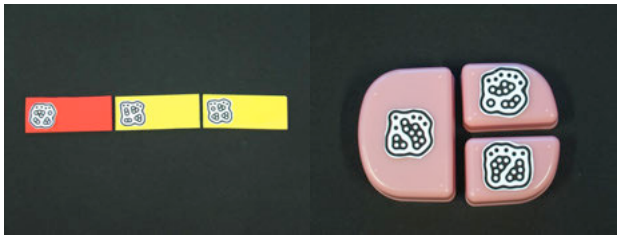


Figure 2. Two examples of chains.

5.1 Master/Slave Objects

Let's imagine a situation where an object would represent a track and a second object a clip to play in this track. The two objects do not have the same hierarchical level since it is the second object that must inherit the Sequence ID from the first one. For this reason, objects have an additional attribute in order to determine if they are Master objects or Slave objects. Master objects will hold a Sequence ID, and slave objects will inherit this Sequence ID when they enter in relation with the master object. If a chain is formed only with slave objects, then they will not get any Sequence ID. Similarly, they will not get an Ordering number since their position in the chain is calculated respectively to the master object (Figure 3, top). Master objects cannot enter in relation, even if their selective filters would allow it. But it can happen that a slave object would relate with more than one master object (Figure 3, bottom). In this case, there are rules of exclusion based either on the order of occurrence (first master-slave relation will supersede any further relation), or on a spatial distribution (for instance, only the leftmost relation will be considered).

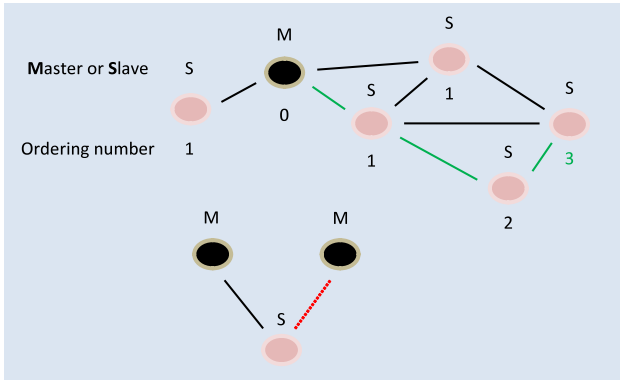


Figure 3. Ordering number attribution (top) and Multiple Slave-Master relation exclusion (bottom).

5.2 Attribution of the Ordering Number

When several objects are in relation in a chain, it is frequent that closed loops are created. In practice, it is better to avoid confusing relationships by setting the appropriate filters between objects. However, we had to find a way to manage any possible situation. For each slave object, the ordering number algorithm searches for a path towards the master object with the shortest distance between neighbors (Figure 3, top). The order is then the number of steps necessary to reach the master object. An additional rule imposes that a direct relation with the master object will supersede any other relation.

Table 1 is giving a summary of all the attributes for an object and if they are defined at the design level (by the user) or at the performance level (by the system).

Table 1. Summary of objects attributes

Attribute	Design or Performance Level
Object ID (= tag ID)	Design
Object size	Design
Type	Design
Tag position	Design
Master or Slave	Design
Sequence ID	Design or Performance
Ordering number	Performance
Chain length	Performance

6. EVENTS

In order to trigger actions when objects are manipulated by users, for instance to trigger loops or to vary sound processing parameters, it is necessary to generate triggering events. They are linked to the input parameters provided by the tracking system (intrinsic parameters and derived parameters), and occur when certain conditions are met. In this study, we consider two families of events, Object Events and Touch Events. We also make a distinction between *discrete* events and *continuous* events present in both families.

6.1 Connect Event

A Connect event is a discrete event that occurs when an object enters in relation with another one. The second object may be either alone or within an already existing chain of objects. The conditions for a Connect event to occur are specified by the filters defined in section 4.4. For instance, an object could generate a Connect event only when it is close enough to the right side of another object of the same type. This would require three filters, one setting the type, one setting the distance, and one setting the range of the valid angle between the two objects.

6.2 Other Object Events

In addition to the Connect event, other object events include:

- Object Down: the object is placed on the surface.
- Object Up: the object is removed from the surface.
- Object Exists: the object is on the surface (continuous).
- Object Moving: position or angle is changing (cont.).
- Object Moved: position or angle has changed (discrete).

6.3 Touch Events

If the tracking system is capable of detecting when objects are being touched, then additional events are available for triggering actions:

- Touch Down: the object is touched.
- Touch Up: the object is stopped being touched.
- Drag Start: a dragging movement starts on the object.
- Drag Stop: the dragging movement stops.
- Touching: lasts while the object is touched (cont.).

Filters are also available in this case for additional selective conditions [6].

7. IMPLEMENTATION

The Surface Editor has been considerably extended in order to support objects management and also to integrate more closely with Ableton Live. A new class of mapping components have been added (Objects), a new activator has been created to handle objects (Object Activator), and two new actions have been added specifically for Ableton (Live Track and Live Device). Also, it is now possible to send variables between objects and controllers in order to change the behavior of an object or controller from another one. This can be used, for instance, to change the MIDI channel of an object's action from the rotation angle of another object. Finally, controllers and objects can be used together for additional flexibility.

Figure 4. Setup.

The Surface Editor receives the touch and object information via TUIO provided by two tracking systems running in parallel. The first one is the Airplane controller developed in previous projects [1], and the other one is a simple webcam hooked to ReacTIVision [10]. Thus, two cameras are looking to the scene (Figure 4, right). A video projector is also used for the optional projection of visual feedback on the table.

Since the Airplane controller is detecting touch by watching fingers crossing a plane of IR light placed a few millimeters above the surface, objects must be thin enough for not interfering with the plane (Figure 5). If interacting with touch gestures is not desired, then thicker objects can be used.

7.1 Linking Controllers to Objects

Controllers rely on a graphical representation, like a fader or keyboard for instance, and need a visual display in order to be manipulated. On the other hand, as mentioned before, objects hold a persistent representation and do not need a display for visual feedback. However, if projecting an image is not an issue, it can be desired to combine the two interaction paradigms. For this reason, several options exist with the Surface Editor. First, it is possible to arrange controllers on a page and leave some blank space in order to use objects at the same time. However, using pages is a rather static approach compared to the manipulation of objects. In order to bring a more dynamic dimension to the use of controllers, it is now possible to link a controller, or group of controllers to an object. Once linked, the controller(s) will appear dynamically when the

corresponding object is placed on the surface (see Figure 5). Moving the object will also move the controller accordingly.

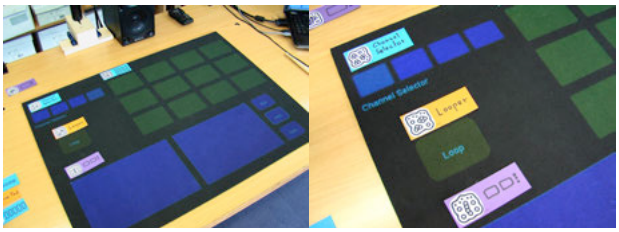


Figure 5. Linking Controllers to Objects.

7.2 Object Activator

The new Object Activator implements the triggering mechanism described in section 6 and the selective filters described in section 4.4. The activator is therefore determining the conditions for an action, or group of actions to occur. Users first select the triggering event and then add as many filters as desired (Figure 6). Like with the Touch Activator available so far with the Surface Editor, several Object Activators can be set for an object, so that several actions or group or actions can be triggered according to different conditions.

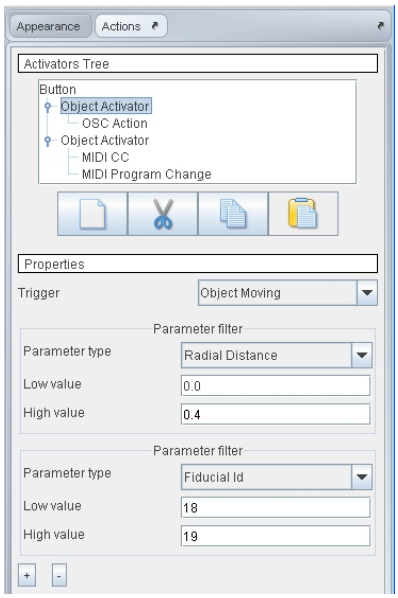


Figure 6. Control panel of an Object Activator.

7.3 Sound Generation

The Surface Editor is now supporting LiveOSC[1] in order to perform bi-directional communication with Ableton Live. This is simplifying enormously the mapping work compared to MIDI. The Surface Editor is informed of the clips, volume, and devices present in a track with all their parameters. Instead of a double work, setting a MIDI CC on one side and setting the same on the other side for the parameter one desires to control, users can map a parameter in Live simply by selecting it in a dropdown list. Two new actions have been created, one to handle track related features (clip, volume, mute, etc.), and one to handle device related features (device parameters).

The Surface Editor is of course not limited to interact with Ableton Live. All variables and information events can be sent through OSC or MIDI to any other sound generation software.

[1] http://liine.net/livecontrol/ableton-liveapi/liveosc/

8. FUTURE WORK

Several user evaluations are planned. First, with novice users using pre-defined scenarios, in order to test the suitability of this system for music pedagogy. Second, the platform will be introduced to more advanced users during a three days workshop that will be held between April 22 and 24 2011, in the context of the Electron Festival[2] in Geneva, Switzerland.

9. ACKNOWLEDGMENTS

The project presented here is supported by the State Secretariat for Education and Research SER, the Swiss National Funding Agency, and the University of Applied Sciences Western Switzerland. Big thanks to Vincent Pezzi for his great work in developping the Surface Editor. Initial work was realized by Pierrick Zoss and Greg Kellum.

10. REFERENCES

[1] Crevoisier, A., and Kellum, G. Transforming Ordinary Surfaces into Multi-touch Controllers. *Proc. of International Conference on New Interfaces for Musical Expression (NIME)*, 2008.

[2] Fitzmaurice, G. W., Ishii, H., and Buxton, W. Bricks: Laying the Foundations for Graspable User Interfaces. *Proc. of Conference on Human in Computing Systems (CHI'95)*, 1995.

[3] Jordà, S. and Kaltenbrunner, M. and Geiger, G. and Bencina, R. The reacTable*. *Proc. of ICMC*, 2005.

[4] Kaltenbrunner, M. http://www.iua.upf.emtg/*reactable*/?related , Referenced October 20, 2006.

[5] Kaltenbrunner, M., Bovermann, T., Bencina, R. and Costanza, E. TUIO - A Protocol for Table Based Tangible User Interfaces. *Proc. of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation* (GW 2005), Vannes (France).

[6] Kellum, G., and Crevoisier, A. A Flexible Mapping Editor for Multi-touch Musical Instruments. *Proc. of NIME-09*, Pittsburgh, USA, 2009

[7] Natural User Interface: http://en.wikipedia.org/wiki/Natural_user_interface

[8] Patten, J., Ishii, H., Hines, J., Pangaro, G. Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. Proc. of *CHI'01*, ACM Press, pp.253-260, 2001.

[9] Patten, J., Reht, B., and Ishii, H. Audiopad: A Tag-based Interface for Musical Performance. *Proc of NIME-02*, (2002), 24-26.

[10] ReactiVision: http://reactivision.sourceforge.net/

[11] Ullmer, B., and Ishii, H. Emerging frameworks for tangible user interfaces. IBM Systems Journal 39 (2000), pp. 915–931.

[12] Wanderley, M., and Depalle, P. Gestural Control of Sound Synthesis. *Proc. of the IEEE*, vol. 92, No. 4 (April), Special Issue on Engineering and Music - Supervisory Control and Auditory Communication, G. Johannsen, Ed., pp. 632-644.

[13] Wolf, M. Soundgarten: A Tangible Interface that Enables Children to Record, Modify and Arrange Sound Samples in a Playful Way. *Masters thesis*, University of Applied Sciences Cologne, Germany, 2002.

11. ADDITIONAL RESOURCES

More info and videos at: www.future-intruments.net

[2] http://www.electronfestival.ch/

Adding Z-Depth and Pressure Expressivity to Tangible Tabletop Surfaces

Jordan Hochenbaum[1]
New Zealand School of Music[1]
PO Box 2332
Wellington, New Zealand
hochenjord@myvuw.ac.nz

Ajay Kapur[1, 2]
California Institute of the Arts[2]
24700 McBean Parkway
Valencia, California, 91355
akapur@calarts.edu

ABSTRACT
This paper presents the SmartFiducial, a wireless tangible object that facilitates additional modes of expressivity for vision-based tabletop surfaces. Using infrared proximity sensing and resistive based force-sensors, the SmartFiducial affords users unique, and highly gestural inputs. Furthermore, the SmartFiducial incorporates additional customizable pushbutton switches. Using XBee radio frequency (RF) wireless transmission, the SmartFiducial establishes bipolar communication with a host computer. This paper describes the design and implementation of the SmartFiducial, as well as an exploratory use in a musical context.

Keywords
Fiducial, Tangible Interface, Multi-touch, Sensors, Gesture, Haptics, Bricktable, Proximity Sensing

1. INTRODUCTION
Musicians have long been intrigued by gestural interfaces since the invention of the Theremin in the early 20th century [2]. This has led to the exploration of pressure-based input sensing for expressive musical interaction. Realizing the potential expressivity of gestural interaction in musical contexts, researches have developed a number of hands-free and pressure based interfaces, exploring several sensing technologies. These include laser controllers such as Hasan, Yu, and Paradiso's work on the the Termenova [3], Wiley's Multi-Laser Gestural Interface [14], Murphy's force-sensing resistor based controller the Helio [10], and countless others.

Concurrently, the last few years has seen an explosion of interest in musical tangible interaction including the Reactable [7], the Bricktable [4, 5], Block Jam [11], and the Audiopad [13]. The Microsoft Secondlight project [6] is an interesting example of adding additional input freedom to tabletop surfaces by alternating projection quickly between two independent diffuse surfaces (the tabletop and ones above the tabletop). While Secondlight can track tangibles and gesture above the surface, it lacks distance tracking. Tangible surfaces can undoubtedly provide users with extremely dynamic interaction, however they lack the gestural qualities of non-contact and pressure based interfaces.

The SmartFiducial is an attempt to provide the best of both worlds—offering and expanding upon the traditional *x,y*, and *rotational* modes of interaction afforded by tabletop surfaces, while providing the gestural expressivity and sensatory affordances experienced from hands free and pressure based interaction.

The remainder of this paper is organized as follows. The Implementation section is divided into two subsections; the first describes the physical design and technology embedded within the SmartFiducial, and the second describes the use of the SmartFiducial with an interactive musical application. The Discussion section details the various design considerations and affordances of the SmartFiducial.

2. IMPLEMENTATION
The SmartFiducial offers users multiple degrees of freedom and expressivity. In this section, we describe the hardware design of the SmartFiducial that enable these input freedoms, as well as our exploratory software implementation of using SmartFiducial's in a musical setting. Figure 1 below provides a general overview of the SmartFiducial tracking system, which is further expounded upon in the following section.

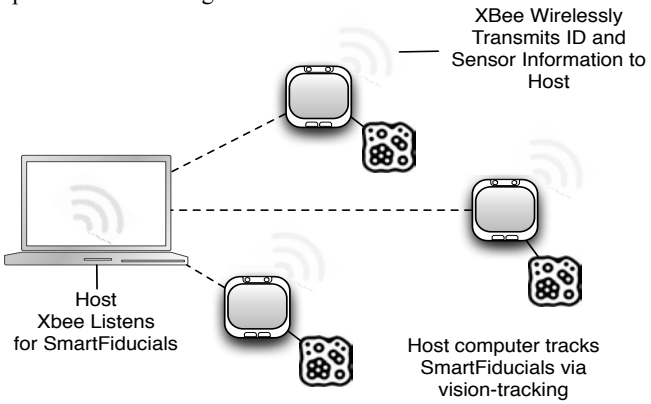


Figure 1 – SmartFiducial System Overview Diagram

2.1 Hardware
2.1.1 Vision Tracking
X, *Y* and *Rotation* tracking is achieved using a custom version of the open-source vision tracking software CCV (Community Core Vision). CCV implements the *libfidtrack* engine developed for the reacTIVision system [8]. More information on the vision tracking systems communication can be found in section 2.1.5.

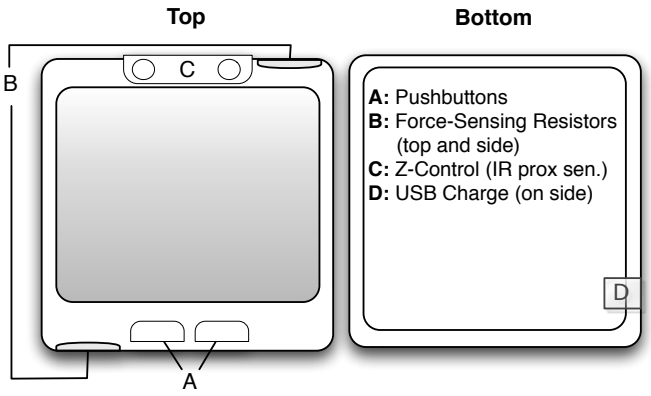


Figure 2: SmartFiducial Hardware Design Layout

2.1.2 Z-Depth

In addition to *x*, *y*, and *rotation* information captured by the vision tracking system, the SmartFiducial enhances the traditional 2D optical tracking system into a three-dimensional space. *Z-depth* input freedom is achieved by a short-range Sharp *GP2D120XJOOF* infrared (IR) proximity sensor embedded on the top face of the SmartFiducial (Figure 2, item C). The *GP2D120XJOOF* has an active sensing range of approximately 3cm – 40cm, providing users with a coverage area capable of highly expressive gesture sensing.

2.1.3 Pressure Sensitivity

The SmartFiducial also provides pressure-based gestural input via two Force-Sensing Resistors (FSR's), on the sides of the SmartFiducial, as pictured in Figure 2, item B.

2.1.4 Wireless Transmission

Embedded within the SmartFiducial is an Arduino Funnel IO[1] (Fio) equipped with an XBee wireless transmission module. XBee utilizes the ZigBee communication protocol, operates at 2.4GHz radio frequency, and exhibits extremely low power-consumption properties. This makes XBee an excellent candidate for wireless serial communication between the SmartFiducial and a host computer. Additionally, the XBee provides each SmartFiducial with a unique identifier, tied to its fiducial ID.

Data is received wirelessly via and XBee connected to the host machine, and is parsed by our custom version of CCV. CCV then sends out the ID and sensor data to other client applications using a custom implementation of the TUIO[2] protocol [9] that supports our additional data.

`/tuio/smartFid set sId id x y z a X Y A m r f F b B`

sId	Session ID	int32
id	Fiducial ID	int32
x, y, z	Position	float32
a	Angle	float32
X, Y	Velocity Vector (motion speed & direction)	float32
A	Rotation velocity vector (rotation speed & direction)	float32
m	Motion Acceleration	float32
r	Rotation Acceleration	float32
f, F	Pressure	float32
b, B	Button-state	int32

Figure 3: SmartFiducial TUIO Protocol Specification

[1] The Arduino Funnel IO is an Atmega based microprocessor designed by Shigeru Kobayash

[2] TUIO is a UDP based data-communication protocol, built around Open Sound Control (OSC) [15]

Additionally, we have implemented a basic algorithm in the SmartFiducial firmware to only broadcast new data when input is detected. This optimization helps to reduce the amount of data being transferred in larger system use-cases and scenarios, and can optionally be turned off in the firmware if constant streaming is preferred.

Once CCV receives new data bundles from the connected Xbee, it first checks to make sure that the SmartFiducials ID is present in the list of active fiducials being tracked by the vision system, before broadcasting a new TUIO message. This prevents the SmartFiducials sensor data from being transmitted when not active on the tabletop surface, however, this can optionally be turned off if off-surface interaction is desired. Although the SmartFiducial messages include all information present in standard TUIO fiducial ("/2Dobj") messages, CCV also broadcasts the SmartFiducial as part of its regular fiducial message broadcasting. Lastly, support for the SmartFiducial has been added into the standard C++ TUIO client implementation allowing easy integration into custom software applications. We are currently working to include support for the SmartFiducial in other TUIO client implementations (Java, Processing, openFrameworks, Max/MSP, Pure Data, etc) however, the SmartFiducial data can still be accessed cross-platform via any OSC receiver application or library.

2.1.5 Serial Protocol

Figure 4 below outlines the serial-protocol developed for SmartFiducial communication. All data is sent to the vision tracking software in 6-byte message bundles.

Fiducial ID has a resolution of 8-bits, yielding support for 255 unique fiducial IDs. All analog sensors (*Z-depth* and *Pressure Sensitivity*) retain full 10-bit resolution, while digital inputs (buttons) use 1-bit respectively. An additional 2-bits (bits 0 and 1 in byte5) are reserved for two additional digital sensors in the future. Lastly, the most significant bit (MSB) in each of the six-bytes is reserved as a special alignment bit, which is checked in CCV in order to ensure robustness and reliability of the wireless serial communication.

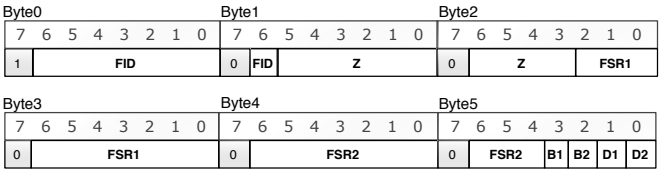


Figure 4: Overview of the SmartFiducial Serial Protocol

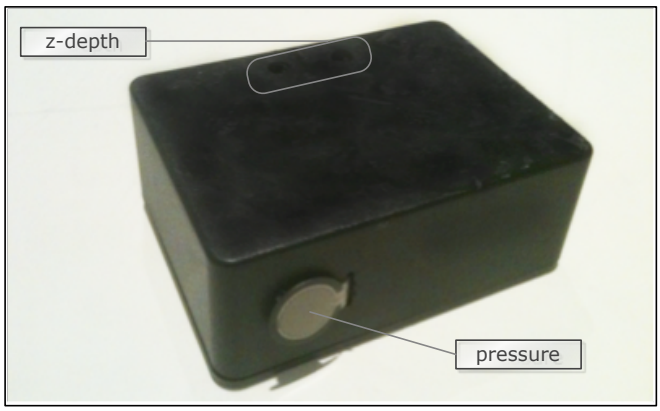


Figure 5: SmartFiducial Prototype (buttons 1 & 2 not pictured)

2.2 Software

In order to begin exploring the unique interactions afforded by the SmartFiducial, we have developed a basic wavetable synthesizer sequencer called *Turbine* (Figure 6). When a SmartFiducial is placed on the tabletop surface, sixteen "nodes" are created around the object. Each node represents a sixteenth note in a one-bar sequence, and dragging the node away from the SmartFiducial changes the pitch of the step. Using the z-depth sensing, the user is able to gesturally morph between the wavetables single-cycle waveforms, creating highly expressive, complex oscillations. Visual feedback is provided to the user via a soft Gaussian circle emitting from underneath the SmartFiducial. Currently the circle grows larger in size as the user nears the SmartFiducials proximity sensor, although the visual feedback may change as we continue to add more functionality to Turbine. In the future, we hope to expand Turbine's functionality, including the interaction between multiple SmartFiducials as well as regular fiducial objects acting as sound modifiers, effects, and other types of intermediaries.

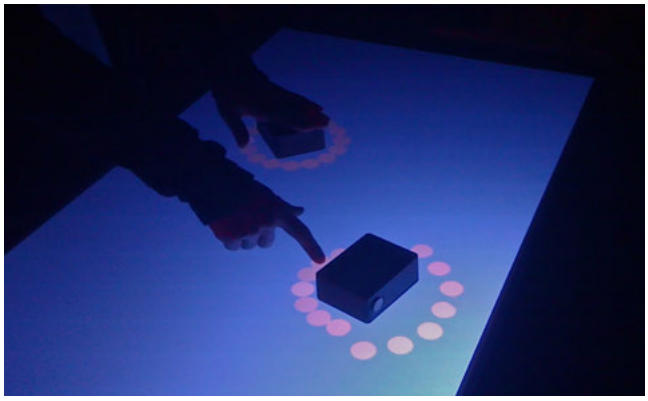


Figure 6: Two SmartFiducials being used with Turbine

3. DISCUSSION

3.1 Spatial Relationships and Tangible Interfaces

Because tangible surface interaction happens along the *xy plane*, input interaction is often a result of 2D spatial manipulation of the objects and the resulting relationships to both the tabletop surface and/or other objects. Once a tangible is placed within this location-dependent context however, e.g. when actions are tied to specific *xy* coordinates on the surface, the *x* and *y* input freedoms are no longer useable (without changing the set relationships). In this situation, the only user-interaction possible is by rotating the object, or interacting with a virtual parameter displayed on the surface itself (assuming the surface is also touch-enabled). Although manipulating on screen parameters can often be effective for input, it poses many user interface (UI) challenges (clogging the UI, dealing with movable UI elements tied to the fiducials….etc) and is often less than ideal. Additionally, proximity sensing and pressure based input offer a wide range of affordances not possible by other means, as further discussed in the following section. Thus, the addition of z-depth proximity sensing and pressure sensitivity on the SmartFiducial allows tangible interaction to be more expressive in this situation, and other scenarios in the following ways:

- Adding complementary modes of input that can be utilized independently or simultaneously with traditional *x*,*y*, and *rotational* tangible interaction
- Beginning to address the loss of input modes in situations where the object must be placed in specific locations or when *xy* spatial relationships and movement are primary means of surface interaction.

This greatly strengthens the ability of having dynamic relationships possible between tangible objects and the surface, and also between tangible objects and neighboring objects.

3.2 New Affordances for Tabletop Interaction

Affordance theory, originally proposed by perceptual psychology pioneer J.J. Gibson introduces the idea that the potential utility of an object is based on the perceived qualities of the object by the subject [1]. Whereas previous work in vision-based tangible tabletop surfaces has given users a set of interaction affordances defined by spatial relationships within a 2D environment, the SmartFiducial not only extends these affordances into the third dimension, but also offers additional affordances, governed by the unique cognitive notions of gesture based input. The following are a few of the interaction affordances that we have discovered through our initial experimentation with the SmartFiducial:

- Z-Depth proximity sensing may provide a more natural means of exploring 3D virtual environments on tabletop surfaces compared to traditional 2D interfaces.
- Both pressure sensitivity and proximity sensing offer the user new means of highly gestural continuous control. These are very different than common touch-based input gestures such as pinching, zooming…etc
- Pressure sensitivity is not only gestural but may afford the user more tactile interaction and control over traditional tangible interaction, especially when in combination with other interaction techniques (for example, utilizing the pressure sensors simultaneously with moving and/or rotating the objects on the surface).
- Proximity and pressure sensors lend themselves particular well to the application of a parameter modifier, non-dependent on the tabletop surfaces GUI

Additionally, the design of the SmartFiducial is influenced by Donald Normans application of Affordance Theory to the field of Design, and Human Computer Interaction (HCI) [12]. In accord with Normans idea that the design of an object can be such that it suggests potential usage, our qualitative use of SmartFiducials has matured in its design in ways we believe optimize the SmartFiducial to be naturally used, without previous experience. This includes the interaction design decision to place the IR proximity sensor on the top of the SmartFiducial, and the pressure sensors both on the sides of the SmartFiducial, typically where users tend to grip the object. While of course there will always be a familiarization stage between the user and the software running on the tabletop surface, our initial exploratory testing showed that when the users knew there was a distance sensor on the top and pressure sensors on the sides, they were able to very naturally exert a high-level of control and nuance in the use of the inputs.

4. CONCLUSION

Building upon previous vision-based tangible surface interaction techniques (offering *x*,*y* and *rotational* modes of input freedoms), the SmartFiducial is a novel tangible object which offers a new level of gesture and tactile affordances to tangible tabletop interaction. While we present an initial exploratory application of these new input freedoms in the creative music realm (Turbine), we believe the potentials afforded by the SmartFiducial can greatly enhance the user-experience when interacting with tangible tabletop surfaces across many disciplines and fields.

We are currently developing the Turbine synthesis engine to more thoroughly examine the affordances of the SmartFiducial in musical contexts. In the future we are particularly interested in conducting user-studies that explore our preliminary findings and experiences with the SmartFiducial (sections 3.1 and 3.2), and will also hopefully illuminate new use cases and affordances of the SmartFiducial.

Additionally, we are excited to finally release the SmartFiducial and our branch of CCV out into the community and see how others interpret and apply the new input freedoms.

5. ACKNOWLEDGMENTS

Special thanks to Owen Vallis, Jim Murphy, and Dimitri Diakopoulos.

6. REFERENCES

[1] Gibson, J. *The Ecological Approach To Visual Perception*. Lawrence Erlbaum Associates, 1986.

[2] Glinsky, A. *Theremin: Ether Music and Espionage*. University of Illinois Press, 2000.

[3] Hasan, L., Yu, N. and Paradiso, J.A. The Termenova: a hybrid free-gesture interface *Proceedings of the 2002 conference on New interfaces for musical expression*, National University of Singapore, Dublin, Ireland, 2002.

[4] Hochenbaum, J. and Vallis, O., Bricktable: A Musical Tangible Multi-Touch Interface. in *Proceedings of Berlin Open Converence 09'*, (Berlin, Germany, 2009).

[5] Hochenbaum, J., Vallis, O., Diakopoulos, D., Murphy, J. and Kapur, A., Designing Expressive Musical Interfaces for Tabletop Surfaces. in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (Sydney, Australia, June 2010).

[6] Izadi, S., Hodges, S., Taylor, S., Rosenfeld, D., Villar, N., Butler, A. and Westhues, J., Going beyond the display: a surface technology with an electronically switchable diffuser. in *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, (Monterey, CA, USA, 2008), ACM, 269-278.

[7] Jordà, S., Kaltenbrunner, M., Geiger, G. and Bencina, R., The reacTable. in *Proceedings of the International Computer Music Conference* (Barcelona, Spain, 2005).

[8] Kaltenbrunner, M. and Bencina, R. reacTIVision: a computer-vision framework for table-based tangible interaction *Proceedings of the 1st international conference on Tangible and embedded interaction*, ACM, Baton Rouge, Louisiana, 2007.

[9] Kaltenbrunner, M., Bovermann, T., Bencina, R. and Costanza, E., TUIO - A Protocol for Table Based Tangible User Interfaces. in *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation* (2005).

[10] Murphy, J., Kapur, A. and Burgin, C., The Helio: A Study of Membrane Potentiometers and Long Force Sensing Resistors. in *in Proceedings of the International Conference on New Interfaces for Musical Expression*, (Sydney, Australia, June 2010).

[11] Newton-Dunn, H., Nakano, H. and Gibson, J. Block jam: a tangible interface for interactive music *Proceedings of the 2003 conference on New interfaces for musical expression*, National University of Singapore, Montreal, Quebec, Canada, 2003.

[12] Norman, D. *The Psychology Of Everyday Things*. Basic Books, 1988.

[13] Patten, J., Recht, B. and Ishii, H. Audiopad: a tag-based interface for musical performance *Proceedings of the 2002 conference on New interfaces for musical expression*, National University of Singapore, Dublin, Ireland, 2002.

[14] Wiley, M. and Kapur, A., Multi-Laser Gestural Interface - Solutions for Cost-Effective and Open Source Controllers. in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (2009).

[15] Wright, M., Freed, A. and Momeni, A. OpenSound Control: state of the art 2003 *Proceedings of the 2003 conference on New interfaces for musical expression*, National University of Singapore, Montreal, Quebec, Canada, 2003.

Hex Player—A Virtual Musical Controller

Andrew J. Milne
Computing Department
The Open University
Milton Keynes, UK
andymilne@tonalcentre.org

Anna Xambó
Computing Department
The Open University
Milton Keynes, UK
a.xambo@open.ac.uk

Robin Laney
Computing Department
The Open University
Milton Keynes, UK
r.c.laney@open.ac.uk

David B. Sharp
Department of DDEM
The Open University
Milton Keynes, UK
d.sharp@open.ac.uk

Anthony Prechtl
Independent Researcher
California, USA
aprechtl@gmail.com

Simon Holland
Computing Department
The Open University
Milton Keynes, UK
s.holland@open.ac.uk

ABSTRACT

In this paper, we describe a playable musical interface for tablets and multi-touch tables. The interface is a generalized keyboard, inspired by the Thummer, and consists of an array of virtual buttons. On a generalized keyboard, any given interval always has the same shape (and therefore fingering); furthermore, the fingering is consistent over a broad range of tunings. Compared to a physical generalized keyboard, a virtual version has some advantages—notably, that the spatial location of the buttons can be transformed by shears and rotations, and their colouring can be changed to reflect their musical function in different scales.

We exploit these flexibilities to facilitate the playing not just of conventional Western scales but also a wide variety of microtonal generalized diatonic scales known as moment of symmetry, or well-formed, scales. A user can choose such a scale, and the buttons are automatically arranged so their spatial height corresponds to their pitch, and buttons an octave apart are always vertically above each other. Furthermore, the most numerous scale steps run along rows, while buttons within the scale are light-coloured, and those outside are dark or removed.

These features can aid beginners; for example, the chosen scale might be the diatonic, in which case the piano's familiar white and black colouring of the seven diatonic and five chromatic notes is used, but only one scale fingering need ever be learned (unlike a piano where every key needs a different fingering). Alternatively, it can assist advanced composers and musicians seeking to explore the universe of unfamiliar microtonal scales.

Keywords

generalized keyboard, isomorphic layout, multi-touch surface, tablet, musical interface design, iPad, microtonality

1. INTRODUCTION

Hex Player is a virtual musical controller. It is played by the fingers and sends standard MIDI messages to control any software or hardware synthesizer. It is designed to make playing music easier without imposing a ceiling on expressiveness and creative possibilities. Hex Player lives on a multi-touch surface such as a tablet (e.g., iPad) or table (e.g., Evoluce or Microsoft Surface), and consists of a lattice (array) of hexagonal buttons (see Figure 1).

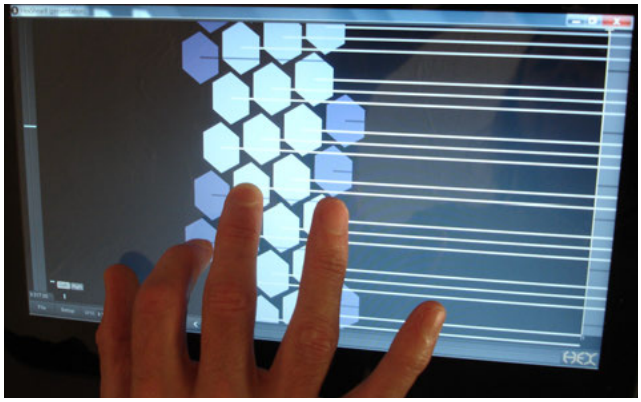


Figure 1: Hex Player on a tablet. The white/grey buttons are generalized diatonic/chromatic notes.

Pressing any of these virtual buttons sends a MIDI note event to a software, or external hardware, synthesizer. The blank areas to the left and right of the buttons are a control surface that can be operated by the thumb or pinkie (little finger) of each hand; these can be used to control a number of expressive parameters such as timbre, volume, vibrato, or tuning, while the four fingers (or three fingers and thumb) play notes. This means that an expressive lead part (with pitch bends, and vibrato, etc.) can be played with one hand, and a bass line or chords with the other. Contrast this with a piano-style keyboard, where an expressive lead part typically requires two hands—one to play the notes, the other to operate the pitch-bend/modulation wheel/joystick.

Hex Player's note layouts and use of a thumb-operated controller are based upon the design of Thumtronic Inc.'s prototypical hardware instrument, the Thummer (`http://thummer.com`). The Thummer project is now open source and, in this paper, we describe how we have transferred many of the design features of the hardware Thummer to a software virtual interface, and how we have further extended its capabilities in ways that would not be possible in a hardware device.

Like the Thummer, Hex Player provides a generalized keyboard and uses an isomorphic note layout. Generalized keyboards have their keys, or buttons, arranged in a regular (typically two-dimensional) lattice. Such keyboards date back at least as far as the nineteenth century; many exam-

NIME'11, 30 May–1 June 2011, Oslo, Norway.

ples of which are given in [6]. On a generalized keyboard, it is possible to arrange pitches isomorphically, which ensures that the same interval, scale, or chord, always has the same geometrical shape regardless of transposition [2, 7]. Furthermore, intervals, scales, and chords (when categorized according to reasonable criteria) have the same geometrical shape over a wide range of tunings, thus enabling tuning to be altered without affecting note layout [8].

In this paper, we describe four novel extensions to the current theory of generalized keyboards and isomorphic note layouts: firstly, we introduce a class of *adjacent seconds* note layouts, which generalize many of the useful properties of the Wicki accordion button layout (the layout used by the Thummer) over a wide variety of microtonal scales; secondly, we describe how shears and rotations of the layouts ensure the pitch height of each button is shown by its spatial height, and that buttons an octave apart are vertically aligned; thirdly, we show how the amount of shear applied to the layout (and hence its tuning) can be dynamically controlled while playing; fourthly, we show how alterable button colouration can be used to indicate generalized diatonic and chromatic scales (well-formed or MOS scales—formally defined in Section 2.1.4).

In the next section, we provide some of the music theoretical and mathematical underpinnings required to understand these innovations. After that, we discuss some of the potential benefits of an interface like Hex Player.

2. GENERALIZED KEYBOARDS AND ISOMORPHIC NOTE LAYOUTS

Generalized keyboards with isomorphic layouts have a number of properties that may facilitate the comprehension and playing of music.

2.1 Basic Definitions

Before proceeding to a full description of the interface and its properties, some formal definitions may be helpful.

2.1.1 Generalized Keyboard

As pictured in Figure 2, a *generalized* keyboard or *button-lattice* consists of a regular array of (real or virtual) buttons [7], each of which plays a musical tone. The buttons could be arranged in one-, two-, or three-dimensional space, but we will restrict this discussion to two-dimensional lattices (because these can be implemented on a surface).

2.1.2 Two-dimensional Tuning System

A *two-dimensional tuning system* is one that is generated from two intervals—a *period* (typically the octave), and a *generator* (typically a perfect fifth). For example, the pentatonic scale can be generated by stacking four perfect fifths (e.g., C–G–D–A–E) and reducing them by octaves so all tones lie within one octave (e.g., in pitch order, C, D, E, G, A); the diatonic scale can be generated by stacking six perfect fifths (F–C–G–D–A–E–B) and reducing them by octaves (e.g., in pitch order, C, D, E, F, G, A, B); the chromatic scale by stacking eleven such fifths (e.g., E♭–B♭–F–C–G–D–A–E–B–F♯–C♯–G♯) and reducing them (e.g., in pitch order, C, C♯, D, E♭, E, F, F♯, G, G♯, A, B♭, B). The period and generator can, however, take any size (not just octave and fifth), and different sizes can generate unfamiliar scales that share a number of properties with the familiar pentatonic, diatonic, and chromatic.

2.1.3 Isomorphic Layout

An *isomorphic layout* is one in which the period and generator of the tuning system are mapped to a spatial basis of the button-lattice [8]. There are several possible bases, and hence several possible mappings (and each different mapping constitutes a different note layout), but one of the most historically successful (at least in the case of the chromatic scale) can be found in the Wicki accordion button layout [13]. The key to understanding the Wicki mapping is to note how the period and generator (here octave and fifth) are mapped spatially (see Figure 2—the arrows show the spatial vectors corresponding to the period and generator).

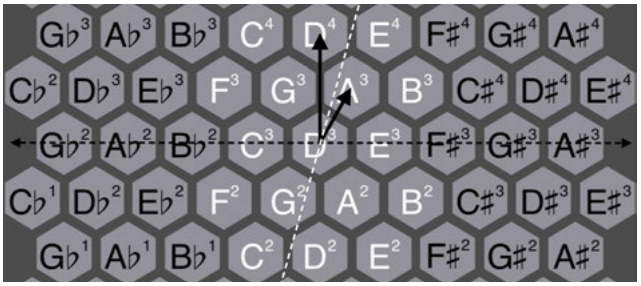


Figure 2: An isomorphic note layout (the Wicki layout) on a virtual generalized keyboard.

2.1.4 Moment of Symmetry/Well-formed Scales

A *moment of symmetry* (MOS) [14] or *well-formed* [4] scale is a generated scale containing exactly two step sizes that are distributed with maximal evenness. A *generated scale* is produced by repeatedly adding a generator interval (typically a perfect fifth) and then reducing all such intervals by repeatedly subtracting a period interval (typically the octave) so all intervals are smaller than the period.

The number of times the generator can be stacked so as to produce just two step sizes depends on the ratio of the generator and period. For example, if the generator is 702 cents and the period is 1200 cents (a generator/period ratio of 0.585), then well-formed scales are available with 3 tones (e.g., C, D, G), 5 tones (e.g., C, D, E, G, A), 7 tones (e.g., C, D, E, F♯, G, A, B), 12 tones (e.g., C, C♯, D, D♯, E, E♯, F♯, G, G♯, A, A♯, B), 17 tones, and so forth [4]. A different generator/period ratio requires different numbers of tones to produce a well-formed scale—when the generator is 316 cents (a just intonation "minor third") and the period is 1200 cents, the following numbers of tones are well-formed: 3, 4, 7, 11, 15, 19, and so forth. With no loss of generality, whenever the size of the period is not explicitly mentioned, it is assumed to be 1200 cents.

MOS scales have a number of properties that are thought to give them æsthetic value. There is not space to discuss these properties in depth but, briefly: every scale span—generic interval size—comes in exactly two specific interval sizes (Myhill's property [5]); the two scale step sizes are evenly distributed throughout the period; within the period, every scale degree has a unique pattern of intervals surrounding it [1]—this may be necessary for any scale to support tonal functionality; when transposed by the generator, the resulting scale shares all but one tone, facilitating modulation [1]; collectively, these features suggest a good compromise between the excessive simplicity of equal step scales and the complexity of completely irregular scales [3].

An MOS, therefore, provides an effective way to choose a scale (a set of notes) from any abstract 2-D tuning.

2.2 General Properties of Isomorphic Layouts

An isomorphic layout has a number of musically useful properties discussed in the subsections below.

2.2.1 Transpositional Invariance

Any given interval, chord, or scale, has the same geometrical shape (and hence fingering) regardless of its location (transposition) on the keyboard. So, a musician need learn the fingering for a major scale, or harmonic minor scale, or third inversion of a dominant seventh chord, just once, and then apply that same shape to any key or root note. Compare this to the piano, where every different major scale requires a different pattern of notes to be memorized [8].

2.2.2 Tuning Invariance

The fingering of a wide range of scales can remain invariant over a wide range of tunings. For example, the fingering of diatonic/chromatic music will stay essentially invariant when the generator has any size between 685.714 and 720 cents. This continuum of tunings includes many notable tunings, such as 19-tone equal temperament (19-TET), various meantone tunings, 12-TET, Pythagorean, 17-TET and 22-TET, so consistent fingering may facilitate the exploration of alternative tunings [8]. There are many different tuning continua, each of which smoothly connects a wide variety of notable tunings [9].

2.2.3 Pitch Axis

Any isomorphic layout has a pitch axis, and the position of any button centre in relation to this axis indicates its pitch. The angle of this axis depends on the note layout (spatial mapping of the period and generator) used and the tuning ratio of the generator and period [9]. For example, in Figure 2, the pitch axis for a generator of 700 cents (which corresponds to a 12-TET tuning) is identified by the dashed white line—note how the distance, measured along this line, between D3 and E3 is two thirds of the distance between D3 and F3 (draw three lines between the pitch axis and D3, E3, and F3, such that all three lines are orthogonal to the pitch axis; the distance between the D3 and E3 lines is two thirds the distance between the D3 and F3 lines).

2.2.4 Generator-Span Axis

The generator-span axis is a novel concept introduced in this paper. Any isomorphic layout also has a generator-span axis, such that the distance between any two button centres, as measured on this axis, indicates the number of generators between them. For example, in Figure 2 the generator-span axis is shown with a black dashed line—the distance, measured on this axis, between D3 and A3 is half the distance of D3 and E3 (there is one fifth between the former, and two fifths between the latter).

Assuming the period and generator are linearly independent, any two notes a period (octave) apart have zero generator distance, hence the generator-span axis is orthogonal to the spatial mapping of the period.

Pitch distance and generator distance are two important musical metrics. The importance of the former is obvious; one importance of the latter is that any given MOS scale always forms a strip running parallel with the octaves.

3. HEX PLAYER

In Hex Player, the pitch and generator-span axes are oriented, and a specific isomorphic layout is selected, so as to maximize certain useful criteria described below.

3.1 Orthogonal Axes

Due to their importance, we have endeavoured to make the pitch and generator-span axes easy to discern, visually and haptically. To do this, we use a novel approach: applying shear and rotation transforms of the lattice to make the pitch axis vertical and the generator-span axis horizontal, regardless of the isomorphic layout or tuning being used. The vertical pitch axis means it is easy to know, in advance of playing, the pitch of any button and that, as the hand plays an ascending scale, it moves gradually away from the body. Furthermore, the horizontal generator-span axis ensures that notes an octave apart are vertically aligned.

3.2 Adjacent Seconds Layouts

Adjacent seconds layouts are a novel concept, and generalize, for any possible MOS scale, some of the useful features of the Wicki layout: a notable property of the Wicki layout, when used for the pentatonic and diatonic scales, is that the most numerous scale step (the whole-tone) runs along each row (in Figure 2, observe the scale runs C-D-E and F-G-A-B), while the least numerous scale step ("minor third" in the pentatonic, and semitone in the diatonic) is reached by a "carriage return" skip up to the next row (in Figure 2, observe the steps E-F and B-C).

For the Wicki layout, this neat property breaks down for most other MOS scales. For example, Figure 3a shows the Wicki layout of the MOS scale with 4 large and 7 small steps (the generator is 320 cents)—the most numerous seconds now skip buttons, the scale's pattern is more difficult to make out, and is not spatially compact. However, by choosing an appropriate isomorphic layout, it is possible for any MOS scale to have the adjacent seconds property. Indeed, for the 120 different possible MOS scales with nineteen or fewer tones, only 13 different layouts are required.

Figure 3b shows how an adjacent seconds layout for the 4 large 7 small MOS scale gives a far more compact and easy to understand layout than the Wicki. In Hex Player, a user can first choose an MOS scale (by inputting the number of large and small steps in the scale), and then click on "Optimize Layout" to switch to an adjacent seconds layout.

3.3 Dynamic Tuning

The horizontal generator-span axis ensures all MOS scales form a vertical strip of buttons. The space on either side can, therefore, be used as a control surface accessible to the thumb and pinkie; either of these fingers can change the shear of the lattice, and send a correlated MIDI CC value, while the remaining fingers are playing. The CC message can ensure the synth's tuning matches that implied by the lattice's shear.[1] This opens up the possibility of players dynamically mimicking the expressive intonations used by advanced string and aerophone players. For example, the thumb can be used to move smoothly between meantone tunings that are ideal for sustained chords, and Pythagorean tunings suitable for expressive melodies [11].

Outside of these familiar Western tunings, the performer can move dynamically, and smoothly, to non-Western tunings such as 5-TET (as used in Indonesian slendro [12]) or 7-TET (as used in traditional Thai music [10]).

These large thumb-/pinkie-operated control surface areas can also be mapped to any parameter, so may also control other pitch or timbral features (e.g., vibrato or brightness).

3.4 Generalized Chromatic Embeddings

Any MOS scale with m large steps and n small can be embedded in an MOS scale with $2m+n$ steps. For example, the pentatonic scale can be embedded within the diatonic scale, which can be embedded within the chromatic, and so on [4]. This provides a neat method to generalize the

[1]Dynamic tuning changes such as these require the use of a Dynamic Tonality synthesizer such as TransFormSynth, The Viking, and 2032, which can be downloaded from `http://www.dynamictonality.com`.

Figure 3: The scale degrees of the 4 large, 7 small steps MOS scale (the generator is 320 cents) in a) the Wicki layout, and b) an adjacent seconds layout. The horizontal lines show the pitch height of every button.

"chromatic" embedding of any MOS scale. In Hex Player, when an MOS scale is chosen (by entering the number of large and small steps), it is automatically displayed as light-coloured buttons (in a central vertical band), surrounded by dark-coloured "chromatic" notes.

This allows for scales of increasing complexity to be presented to a beginner in a consistent fashion. For example, children are frequently taught the pentatonic scale (e.g., C, D, E, G, A) as a first step in their musical education (e.g., the Orff and Kodály methods). In Hex Player, the pentatonic scale can be shown as a light coloured vertical strip of buttons, while the more challenging diatonic notes (in this case, F and B) are dark-coloured buttons on either side of this strip. When the student is ready, the diatonic scale can be shown as vertical strip of light-coloured buttons, with the more challenging chromatic tones shown in a dark colour either side. All of these representations are consistent—the same spatial shape always plays the same interval—but, as the scales become more complex, the strip just gets wider.

4. CONCLUSIONS

This paper presents a novel virtual musical interface designed to facilitate the playing of both familiar and unfamiliar musics by musicians at all levels: from beginners to advanced microtonalists. We do this by combining many of the well-established advantages of generalized keyboards with some novel extensions.

In summary, the interface allows a user to select any MOS scale, and its tuning, such that: a) all intervals have the same shape regardless of transposition; b) all intervals (categorized by reasonable criteria) have the same shape over a wide range of tunings; c) the layout ensures the pitch axis and generator-span axis are vertical and horizontal, respectively, and so are visually and haptically salient and distinct; d) the most numerous scale steps run along rows, the less numerous are "carriage returns" up to the next row; e) the notes in the scale and the "chromatic" scale, within which it is embedded, are displayed as light- and dark-coloured buttons lying in a vertical strip at the centre of the display; f) the space either side of the note strip can be used as a control surface, enabling the thumb or pinkie to dynamically alter the tuning (and correspondingly change the shear of the lattice so as to reflect the resulting pitches of the buttons) while the remaining fingers play.

The dynamically changing shears, rotations, and button colourings are difficult to implement in a hardware interface, hence the usefulness of the virtual realisations made possible by multi-touch surfaces. A drawback of current surfaces, however, is the lack of tactile feedback and velocity and pressure sensitivity. It is unknown, at this stage, to what extent this impacts upon their utility; we intend to explore these issues in future research.

5. ACKNOWLEDGMENTS

Thanks to Jim Plamondon for the original inspiration.

6. REFERENCES

[1] G. J. Balzano. The pitch set as a level of description for studying musical perception. In M. Clynes, editor, *Music, Mind, and Brain*. Plenum Press, New York, 1982.

[2] R. H. M. Bosanquet. *Elementary Treatise on Musical Intervals and Temperament*. Macmillan, London, 1877.

[3] N. Carey. Coherence and sameness in well-formed and pairwise well-formed scales. *Journal of Mathematics and Music*, 1(2):79–98, 2007.

[4] N. Carey and D. Clampitt. Aspects of well-formed scales. *Music Theory Spectrum*, 11(2):187–206, 1989.

[5] J. Clough, N. Engebretsen, and J. Kochavi. Scales, sets, and interval cycles: A taxonomy. *Music Theory Spectrum*, 21(1):74–104, 1999.

[6] H. L. F. Helmholtz. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Dover, New York, 1877.

[7] D. Keislar. History and principles of microtonal keyboards. *Computer Music Journal*, 11(1):18–28, 1987.

[8] A. J. Milne, W. A. Sethares, and J. Plamondon. Isomorphic controllers and Dynamic Tuning: Invariant fingering over a tuning continuum. *Computer Music Journal*, 31(4):15–32, 2007.

[9] A. J. Milne, W. A. Sethares, and J. Plamondon. Tuning continua and keyboard layouts. *Journal of Mathematics and Music*, 2(1):1–19, 2008.

[10] D. Morton. The music of Thailand. In E. May, editor, *Music of Many Cultures*, pages 63–82. University of California Press, Berkeley, CA, 1980.

[11] J. Sundberg, A. Friberg, and L. Frydén. Rules for automated performance of ensemble music. *Contemporary Music Review*, 3(1):89–109, 1989.

[12] W. Surjodiningrat, P. J. Sudarjana, and A. Susanto. *Tone Measurements of Outstanding Javanese Gamelan in Yogyakarta and Surakarta*. Gadjah Mada University Press, Yogyakarta, Indonesia, 1993.

[13] K. Wicki. Tastatur für musikinstrumente, Oct. 1896.

[14] E. Wilson. Letter to Chalmers pertaining to moments-of-symmetry/Tanabe cycle, 1975.

Rhythm Performance from a Spectral Point of View

Carl Haakon Waadeland
Department of Music
Norw. Univ. of Science and Tech.
7491 Trondheim, Norway
carl.haakon.waadeland@ntnu.no

ABSTRACT

Basic to both performance and experience of rhythm in music is a connection between musical rhythm and patterns of body movements. A main focus in this study is to investigate possible relations between movement categories and rhythmic expression. An analytical approach to this task is to regard a musician´s various ways of moving when playing an instrument as an expression of timbral aspects of rhythm, and to apply FFT to empirical data of the musician´s movements in order to detect spectral components that are characteristic of the performance.
In the present paper we exemplify this approach by reporting some findings from empirical investigations of jazz drummers´ movements in performances of swing groove. In particular we show that performances of the groove in three different tempi (60, 120, 300 bpm) yield quite different spectral characteristics of the movements. This spectral approach to rhythm performance might suggest alternative ways of constructing syntheses and models of rhythm production, and could also be of interest for the construction of interfaces based on detecting spectral properties of body movements.

Keywords

Rhythm performance, movement, gesture, spectral analysis, swing

1. INTRODUCTION

Most research on rhythm performance has focused on investigating attack point rhythm, i.e. attack points (the temporal start point of the sound) and durations. However, for the performing musician attack points and durations are audible sounding consequences of continuous interactions between the musician, a musical instrument and different physical/ social environments related to which the performance takes place. To obtain a more comprehensive understanding of various interactions of the parameters underlying the characteristics of a rhythm performance it is, therefore, necessary to take gestural aspects of the performance into account in addition to the study of attack point rhythm. It is interesting to know that this point was made already in 1929 by Bernstein and Popova. They carried out empirical investigations of movements in piano playing and stated:"One can say that, with the slow, middle and fastest paces, we are dealing with three totally different dynamical constructions, with 3 dissimilar movement mechanisms" (see [2]). In this paper we discuss to what extent the findings of Bernstein and Popova may be translated from investigations of piano performance to a situation where jazz drummers are playing the swing groove.

1.1 The Swing Groove

A swing groove, as played on a cymbal by a jazz drummer, is often written in the following way:

A jazz drummer may perform a swing groove in a number of different ways. Typically, the drummer will perform the rhythm by making various deviations from the exact note values in the notation above. Moreover, various performances of the swing groove may in varying degree be musical appropriate (more or less "correct" related to various styles or traditions of performance), and may also in varying degree be swinging (i.e. make you want to "swing along with the music").

Interesting studies of performances of swing groove in jazz are presented in [14], [13], [12], [4] and [8]. The main strategy in these investigations has been to measure to what extent live performances show deviations from exact note values, and to relate these deviations to various musical/ contextual parameters, such as individual preferences, musical style, tempo, and inter-ensemble relations (i.e. rhythmic deviations between the different musicians in an ensemble). Several empirical investigations detecting various deviations in performances of music belonging to other traditions than the jazz tradition have also been carried out (e.g. [1], [10], [7], and for and overview, see [3] and [9]).

Studies of gestural aspects of rhythm performance have also been undertaken, although not to the same extent as studies of attack point rhythm. With focus on the drummer´s movements, interesting results have been published by S. Dahl [5] and [6], and the author of this paper has also contributed to research in this direction through empirical studies of jazz drummers´movements (see [16], [17] and [18]). The present report is a continuation of these investigations.

Having established the background for our research, we will now address the following questions:

(a) What information is given by studies of gestural aspects of rhythm performance?
(b) How can spectral analysis of rhythmic movements contribute to new insight into gestural rhythm?

And to be somewhat more specific:

(c) To what extent can the findings of Bernstein and Popova (see above) be supported by analysis of a jazz drummer playing the swing groove?

Parts of these investigations were also reported by the author in an oral presentation at ICMPC, 2006, see [19], but the results here presented have not been published elsewhere.

2. METHOD

The experimental method and the equipment used for the collection of empirical data are both described below, and are well known within empirical movement science.

2.1 Subjects

The subjects participating in this experiment are semi-pro/ professional drummers acquainted with jazz drumming. All of the drummers were, at the time of the experiment, students or teachers at Section of Jazz Education, Department of Music, Norwegian University of Science and Technology (NTNU).

2.2 Task

The subjects were asked to use one drumstick to play the swing groove restricted to various performance conditions. The performance conditions applied for the results to be presented in this paper, were the following:

Play the swing groove as natural as possible:

(i) In tempo 60 bpm (beats pr. minute)
(ii) In tempo 120 bpm
(iii) In tempo 300 bpm

Results related to other performance conditions (e.g. different placement of accent on the beats of the swing groove) are published in [18].

2.3 Equipment

The measurements were carried out assisted by Geir Oterhals, Section of Movement Science, NTNU. Figure 1 illustrates how the experimental situation was constructed. The figure shows the drummer playing on a "force plate" using one drumstick. Markers were placed on the drummer´s arm and on the drumstick.

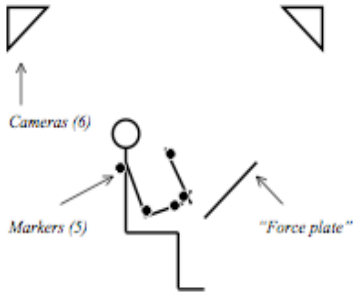


Figure 1. Illustration of experimental set up for measuring kinematic and dynamic aspects of performance of swing groove

For our set up we applied the following components:

- 6 cameras (Proreflex camera system) transmitting infra red light were used to measure movements of the arm and drumstick (kinematics). Sampling frequency = 240 Hz
- 5 markers reflecting the light were placed on the drummer´s shoulder, elbow, wrist and hand, as well as on the tip of the drumstick
- A force plate (Kistler) measuring force from the drumstick was used to give measurements of attack points. Sampling frequency = 960 Hz

3. RESULTS

We now outline some results to demonstrate how the spectral approach to movement analysis might give interesting information that complements insight into gestural aspects of rhythm performance which is achieved from the more commonly used time domain analysis. The results here are taken from a case study that investigates the movements of one drummer playing swing groove in the three different prescribed tempi.

3.1 Analysis in the Time Domain

Figure 2 illustrates the vertical movement (height vs. time) of tip of drumstick, hand and wrist, respectively, of one representative cycle in a performance of swing groove for one drummer in the three tempi: 60 bpm, 120 bpm, 300 bpm.

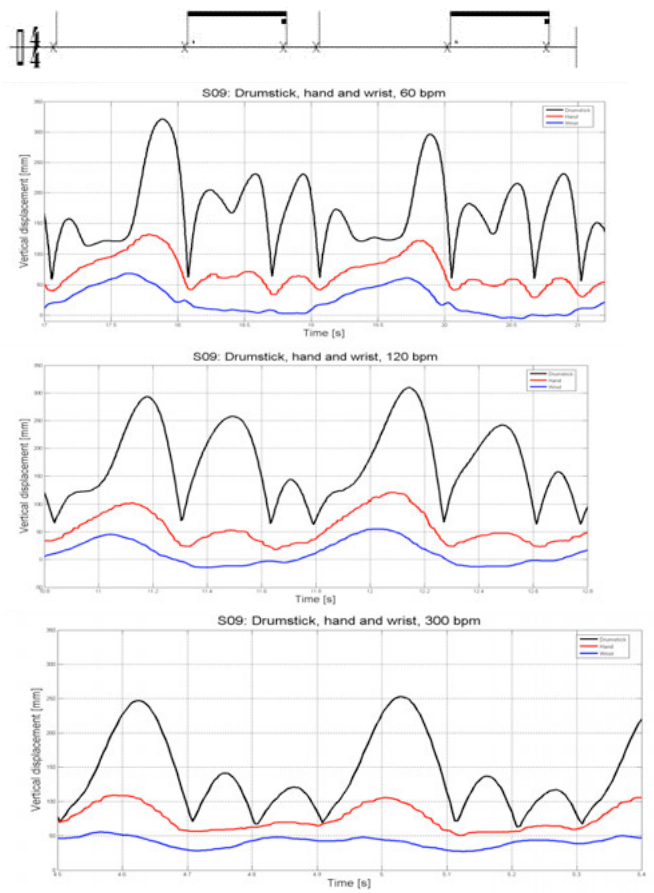


Figure 2. Vertical movement of tip of drumstick, hand and wrist in a performance of swing groove in three different tempi (from top to bottom): 60, 120, 300 bpm

From the illustration in Figure 2 we observe:

In tempo 60 bpm:

- Drumstick "drops" between attack points
- Hand follows every drumstick movement
- Wrist follows the largest hand movement

In tempo 120 bpm:

- No drumstick "drops" between attack points
- Hand follows the two largest drumstick movements
- Wrist follows the largest hand movement

In tempo 300 bpm:

- One large, two smaller drumstick movements in each cycle
- Hand follows the one largest drumstick movement
- Minor movement of wrist

Thus, it seems quite obvious that for this particular drummer, tempo has a major influence on movement patterns and strategy of performance of swing groove.

It is, moreover, interesting to note that tempo also affects the timing of the swing performance. By calculating time

differences in the distribution of attack points in these three performances, we find that the drummer is performing with a triplet-close subdivision at 60 and 120 bpm, whereas the performance approximates eighth note subdivision at 300 bpm (cf. Friberg and Sundström [8] for a similar result). These differences in timing are also reflected in the movement trajectories, as shown in Figure 2.

As indicated above, analysis of gestural aspects of swing performance in the time domain yields interesting information of how tempo influences movements on a local (i.e. time specific) level. Categorical differences in movement patterns are further demonstrated when we examine the global spectral properties of the performances.

3.2 The Spectral Approach

Applying FFT (Fast Fourier Transform) to the movement data of the swing performances, we get the results as shown in Figure 3 (the FFT-analysis was carried out in Matlab):

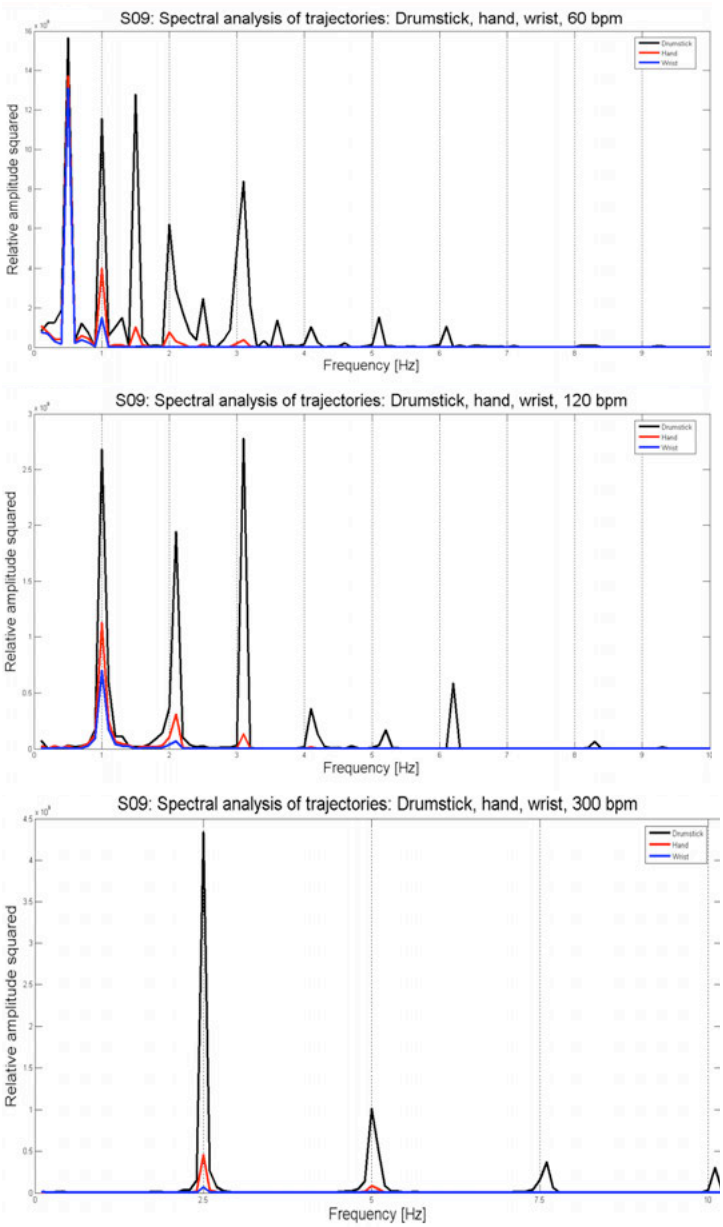


Figure 3. Spectral analysis of movements of swing performance in three different tempi (from top to bottom): 60, 120, 300 bpm

It is interesting to observe that the frequencies of the spectral components in Figure 3 correspond to note values in the different performances of the swing groove. Table 1 shows the relation between note values and frequencies for the three different tempi.

Table 1. Relation between note values and frequencies

	60 bpm	120 bpm	300 bpm
Half note	0,5 Hz	1 Hz	2,5 Hz
Quarter note	1 Hz	2 Hz	5 Hz
Quarter note triplet	1,5 Hz	3 Hz	7,5 Hz
Eight note	2 Hz	4 Hz	10 Hz
Eight note triplet	3 Hz	6 Hz	15 Hz
Sixteenth note	4 Hz	8 Hz	20 Hz

With reference to Table 1, Figure 3 shows:

- In all three performances, the largest spectral components for the hand and wrist movements are given at the frequencies corresponding to the half note (0,5 Hz, 1 Hz, 2,5 Hz resp.). This reflects the fact that the swing pattern is cyclic with a cycle length equal to the duration of a half note, and the movements of the hand and wrist constitute a cyclic-close trajectory.
- One would expect a similar result for the drumstick movements, and, indeed, this is very dominant for tempo 300 bpm, and is also the case for 60 bpm. In tempo 120 bpm, however, the largest component of the drumstick movement for this particular drummer is at 3 Hz, which corresponds to the quarter note triplet.
- In both 60 and 120 bpm the components of drumstick movements corresponding to quarter note triplets and eight note triplets are among the largest in the spectral resolutions. This reflects that our calculation of time differences in the attack point distribution shows that the drummer is performing with a triplet-close subdivision in these two tempi.
- Overall, we see that the number of spectral components decreases with increasing tempo, i.e. when the tempo increases, the movements of drumstick, hand and wrist tend to be more sinusoidal.

We now turn to a discussion of our findings.

4. DISCUSSION

Our approach in the study of gestural aspects of rhythm performance has here been to regard a musician´s movements as an expression of timbral aspects of rhythm, and to apply a combination of analysis in the time domain and spectral analysis of movements. There should be made some comments to the strategy of this study and to the way this experiment was set up:

(1) First of all it should be noted that all together, 10 drummers participated as subjects in this investigation. We have here presented only a case study involving one subject performing swing groove in different tempi. A natural further development would be to study which interactions of performance parameters are common among the drummers. All though it seems plausible that the results here reported show features of performance that are shared among several drummers, these matters should be investigated further in forthcoming studies of swing performance.

(2) In this experiment the drummers were asked to use one drumstick to play the swing groove on a force plate. Obviously, a more natural playing condition would be to ask the drummers to play the groove on a ride cymbal. The main reason for using the force plate is that this performance makes possible a more accurate detection of attack points needed for timing analysis than does a performance on a cymbal. At this point it should be mentioned that some of the subjects participating in the experiment commented that they would feel more comfortable playing the swing groove if they were allowed to use the whole drum set; "filling-in" on the snare, bass drum and hihat underneath the ride cymbal swing pattern. – All this said, it is important to emphasize that we here make comparisons between different performances within the well defined experimental situation, - and we suggest that it is likely that characteristics of various performance parameters that are detected within the experiment, will have validity also in the real "natural" world of music performance – outside the constraints given by the experimental setup.

(3) The study of gestural aspects of rhythm performance is important for the construction of continuous models generating syntheses of rhythm performance that approximate reality. One such model is developed by the author, [15]. It seems likely that results derived from these experiments will be valuable in the further development of this model. – For instance, it might be interesting to reverse the decomposition of movements given by the spectral analysis, in order to simulate rhythm performances on the basis of a given set of sinusoidal components.

(4) Our strategy in the study of swing performance is similar to the approach of Luiz Naveda in analyzing the relationship between samba dancing movement and music, [11]. In our future work it will be investigated to what extent analytical ideas in Naveda´s investigations can be applied to forthcoming studies of swing performance.

The findings reported in this paper show that when a drummer performs the swing groove in different tempi, the tempo is likely to affect the drummer´s movements in various crucial ways. In this respect, our analysis of a jazz drummer playing the swing groove clearly supports the statement made by Bernstein and Popova, [2], cited in 1.Introduction. As a consequence of the findings of Bernstein and Popova, as well as of the results here reported, a performance on the piano (or a performance of swing) in fast tempo is quite different from speeding up a performance on the piano (or a performance of swing) in slow tempo. These findings are likely to be valid for rhythm performance on a more general basis, and should be taken into account in musical training. Moreover, this result is of interest for the constructions of models of rhythm performance.

5. ACKNOWLEDGMENTS

The author is very grateful to Geir Oterhals, Section of Movement Science, NTNU, for fruitful assistance and cooperation in the measurements of rhythmic movements. Special thanks are also due to the drummers participating in the experiments. Moreover, the author would like to thank three anonymous reviewers for valuable comments on an earlier version of this paper. Economical support to this project was provided by The Faculty of Humanities, NTNU.

6. REFERENCES

[1] Alén, O. *Rhythm as duration of sound in Tumba Francesa.* Ethnomusicology, 39(1), 55-71, 1995.

[2] Bernstein, N.A. and Popova, T. *Untersuchung über die Biodynamik des Klavieranschlags.* Arbeitsphysiologie, 1, 396-432, 1929.

[3] Clarke, E.F. Rhythm and timing in music. In *The Psychology of Music, Second Edition,* D. Deutsch (ed.), San Diego (Academic Press), 473-500, 1999.

[4] Collier, G.L. and Collier, J.L. The swing rhythm in jazz. In *Proceedings of the International Conference on Music Perception and Cognition, Montreal,* B. Pennycook and E. Costa-Giomi (eds.), Montreal: McGill University, 477-480, 1996.

[5] Dahl, S. *The playing of an accent – Preliminary observations from temporal and kinematic analysis of percussionists.* Journal of New Music Research, 29(3), 225-234, 2000.

[6] Dahl, S. *Playing the accent – comparing striking velocity and timing in an ostinato rhythm performed by four drummers.* Acta Acustica united with Acustica, 90, 762-776, 2004.

[7] Danielsen, A. *Presence and pleasure – a study of the funk grooves of James Brown and Parliament.* Dr.art.thesis, University of Oslo, 2001.

[8] Friberg, A. and Sundström, A. *Swing ratios and ensemble timing in jazz performance: Evidence for a common rhythmic pattern.* Music Perception, 19(3), 333-349, 2002.

[9] Gabrielsson, A. The performance of music. In *The Psychology of Music, Second Edition,* D. Deutsch (ed.), San Diego (Academic Press), 501-602, 1999.

[10] Kvifte, T. *Fenomenet "asymmetrisk takt" i norsk og svensk folkemusikk.* Studia Musicologica Norvegica, 25, 387-430, 1999.

[11] Naveda, L. *Gesture in Samba. A cross-modal analysis of dance and music from the Afro-Brazilian culture.* Doctoral thesis in Art Sciences. Ghent, Belgium, 2011.

[12] Prögler, J.A. *Searching for swing: Participatory discrepancies in the jazz rhythm section.* Ethnomusicology, 39(1), 21-54, 1995.

[13] Reinholdsson, P. Approaching jazz performances empirically. Some reflections on methods and problems. In *Action and perception in rhythm and music,* A.Gabrielsson (ed.), The Royal Swedish Academy of Music, 55, 105-125, 1987.

[14] Rose, R.F. *An analysis of timing in jazz rhythm section performance.* Unpublished doctoral dissertation. Univiversity of Texas, Austin, 1989.

[15] Waadeland, C.H. *It don´t mean a thing if it ain´t got that swing – Simulating expressive timing by modulated movements.* Journal of New Music Research, 30(1), 23-37, 2001.

[16] Waadeland, C.H. Movements in rhythmic performance of swing in jazz. Report from an empirical investigation. In *Dance Knowledge – Dansekunnskap, Proceedings 6th NOFOD Conference,* Fiskvik and Bakka (eds.), Trondheim, NTNU, 193-199, 2002.

[17] Waadeland, C.H. Analysis of jazz drummers´ movements in performance of swing grooves – a preliminary report. In *Proceedings of Stockholm Music Acoustics Conference 2003 (Vol. II).* R. Bresin (ed.), Stockholm, 573-576, 2003.

[18] Waadeland, C.H. *Strategies in empirical studies of swing groove.* Studia Musicologica Norvegica, 32, 169-191, 2006.

[19] Waadeland, C.H. The influence of tempo on movement and timing in rhythm performance. In *Proceedings of the 9th International Conference on Music Perception and Cognition, Bologna.* M. Baroni, A.R. Addessi, R. Caterina, M. Costa (eds.), Bologna, 29, 2006.

Nuvolet : 3D Gesture-driven Collaborative Audio Mosaicing

Josep M Comajuncosas
Music Technology Group
Universitat Pompeu Fabra
Escola Superior de Música de Catalunya - ESMUC
josep.comajuncosas@esmuc.cat

Alex Barrachina
Escola Superior de Música de Catalunya - ESMUC
alex.barrachina@esmuc.cat

John O'Connell
Music Technology Group
Universitat Pompeu Fabra
johngerardoconnell@gmail.com

Enric Guaus
Escola Superior de Música de Catalunya - ESMUC
enric.guaus@esmuc.cat

ABSTRACT

This research presents a 3D gestural interface for collaborative concatenative sound synthesis and audio mosaicing. Our goal is to improve the communication between the audience and performers by means of an enhanced correlation between gestures and musical outcome. Nuvolet consists of a 3D motion controller coupled to a concatenative synthesis engine. The interface detects and tracks the performers hands in four dimensions (x,y,z,t) and allows them to concurrently explore two or three-dimensional sound cloud representations of the units from the sound corpus, as well as to perform collaborative target-based audio mosaicing. Nuvolet is included in the Esmuc Laptop Orchestra catalog for forthcoming performances.

Keywords

concatenative synthesis, audio mosaicing, open-air interface, gestural controller, musical instrument, 3D

1. INTRODUCTION

Direct manipulation of sound through visual representations, either by gestural, haptic or GUI-based interaction, takes advantage of well established audio representation techniques. By incorporating this paradigm, intuitiveness and easiness of use of such interfaces are maximized. Within this context, content-based navigation and retrieval of audio through scatter plots have become commonplace in MIR-based applications. For instance, Coleman[4] proposes a method for personal sample library exploration based on the analysis of event-synchronous audio segments extracted from a user's digital music collection. Janer[6] presented a sound object browser that allows the user to preview and select the desired target to assign to each step in a looped sequence. Some unit navigation systems for concatenative synthesis and audio mosaicing environments will be reviewed in Section 2.1.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

Nuvolet adds an interaction layer to CataRT[1]. CataRT is a real-time corpus-based concatenative synthesis environment for MaxMSP. It has been developed at the IRCAM by D. Schwarz[15], and allows the user to freely navigate scatter plots of sound corpuses as well as performing target-based audio mosaicing. While there are some cataRT-based systems designed for laptop ensembles (p.ex. Catork[2]), our research is focused on on-stage gestuality to convey expressiveness and intelligibility to the sound exploration process.

Nuvolet bears a strong resemblance to The Enlightened Hands proposed by Viglienson[16]. Comparing both systems, our interface provides unobtrusive multiuser three-dimensional navigation and gestural control of target-based resynthesis.

The paper is organized as follows. In Section 2 we present the most relevant advances in concatenative sound synthesis and in gesture-based interfaces. Next, in Section 3, both the concept and the architecture of the interface are presented, followed by the description of two different case examples, data cloud navigation and interactive target-based mosaicing, in Section 4. Finally, we present a discussion for the system design and interaction issues, and the final conclusions in Sections 5 and 6 respectively.

2. STATE OF THE ART

In this section, we present the most relevant advances in concatenative sound synthesis and in gesture-based interfaces for our work.

2.1 Concatenative Sound Synthesis

Concatenative Sound Synthesis (CSS) is a process whereby audio is created by the concatenation of many small segments of audio, called units, from a source unit database, called a corpus. In this process, unlike in traditional granular synthesis methods, the grain selection is not arbitrary but rather determined by the characteristics of the audio itself. This *data driven process* [14] may take a given audio input as a "target" from which a list of audio features called descriptors are derived.

Source units from the corpus are then selected based on how well they match selected descriptors of the target. Typically, the multi-dimensional descriptor space is searched using a path search algorithm (e.g. Viterbi[14]) or an adaptive local search algorithm (e.g. Zils[20]). This process is called *unit selection.* The target specification is often derived from

[1] `http://imtr.ircam.fr/imtr/CataRT`
[2] `http://www.brunoruviaro.com/catork/`

a piece of audio[20] or from user navigation through the corpus of source units.

Diemo Schwartz has been exploring real time improvisation with CataRT by analyzing and segmenting live audio captured onstage from a musician [3]. Several authors investigate as well how to navigate the multidimensional descriptor space, for example plumage [5], which uses a custom 3D interface to control CataRT. Compared to it, Nuvolet relies on direct mapping from the spatial dimensions to a three-dimensional sound space, thus achieving a touchless but direct manipulation of the virtual timbral space.

2.2 Gesture based interfaces

A pioneering three-dimensional controller was the Radio-Drum, by Boie and M.Mathews, which tracked the batons 3D location by radio-frequency. Another system closer to the interface presented in this paper is Lightning, by D.Buchla, a device which tracked the performer location in the vertical plane by triangulating the infrared transmitters built into baton-like wands. Both were introduced at the early 1990s [3].

The Theremin-like quality of such gestural devices quickly dives into the realms of dance, theater and interactive installations when the space and number of performers increase [10]. Coherently, systems that utilize video capture and IR motion capture devices had been employed since the eighties for dance driven music, as Simon Veitch's 3DIS system [2] and David Rokeby's VNS (Very Nervous System) [18, 19].

More recent developments employ a large number of sensing devices for active location and/or motion capture. A paradigmatic example is the Brain Opera [13], a large multimedia production conceived by Tod Machover and Joseph Paradiso in the late nineties, which implemented a number of open-air sensing techniques, ranging from capacitive sensing for small areas to arrays of ultrasonic range finders or microwave Doopler radars.

Figure 1: Two performers playing the *Nuvolet*, as seen in the monitoring screen

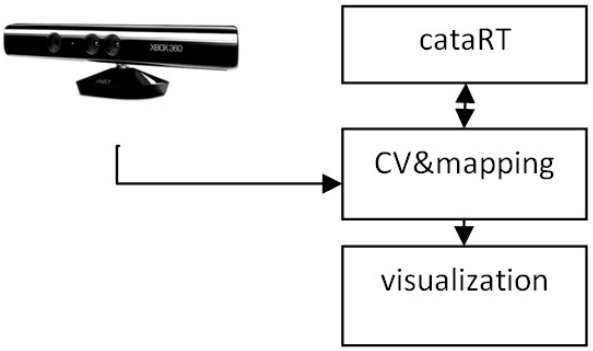


Figure 2: *Nuvolet* block diagram.

3. SYSTEM OVERVIEW

The Nuvolet, originally developed for a musical work written by the catalan composer Ariadna Alsina for singer-reciters and laptop ensemble, is designed to let a number of performers (one to four) of the Esmuc Laptop Orchestra[4] to explore a multidimensional representation of audio snippets by moving their hands in the space.

3.1 Concept

According to the aesthetic intentions of the work, the performers should exemplify some key concepts from the script by visually drawing soundscapes. As the singers evoke places attached to their childhood memories, Nuvolet players navigate a sound corpus made up of field recordings from those locations.

The sound cloud is shown to the performers as a 3D overlay on a monitoring screen, as displayed in Figure 1. Our main goal was to improve the communication with the audience by an enhanced correlation between gestures and musical outcome, achieving at the same time an increased performability compared to the CataRT mouse based GUI. Broad gestures amplify the perception of the player manipulations, which may have a positive effect on the perceived authenticity and expressiveness of the performance.

3.2 System architecture

The system consists of a capture device and a set of software modules, as displayed in Figure 2. A three-dimensional representation of the performer's location is obtained with the Microsoft Kinect[5], which provides an infrared laser projector for robust, ambient light immune depth sensing. The stereoscopic vision is thus achieved through point cloud optical triangulation and the available playing area is about $6m^2$, with a tracking range of 0.7 to 6 meters.

The computer vision software consists of the OpenNI[6] framework, which takes care of the skeleton tracking, and a custom openFrameworks[7] application which performs the required mapping. Only subject and hand tracking were necessary for this project. This application also takes care of the visualization of the sound clouds for performer feedback, as already seen in Figure 1.

Finally, the audio synthesis engine is the concatenative synthesis environment CataRT described in Section 1. Most of the synthesis features required were already available in cataRT, namely audio segmentation, corpus analysis and target driven mosaicing. Only a polyphonic synthesis engine was implemented to minimize interference between performers, and an OSC link was added to interchange data with

[3] http://www.youtube.com/theconcatenator.
[4] http://barcelonalaptoporchestra.blogspot.com/
[5] http://www.xbox.com/es-ES/Xbox360/Accessories/kinect/Home
[6] http://www.openni.org/
[7] http://www.openframeworks.cc/

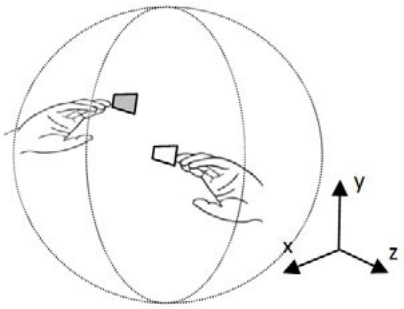


Figure 3: Collaborative sound navigation in Nuvolet.

the visual layer.

4. CASE EXAMPLES

This section shows two different mapping strategies: the first one is designed to concurrently explore two or three-dimensional sound cloud representations of the units from the sound corpus, and the second one is oriented towards collaborative target-based audio mosaicing.

4.1 Descriptor space navigation

In our first scenario, a simple mouse replacement for the CataRT data explorer was implemented, extending the original 2D mouse-driven LCD GUI to a three-dimensional data cloud with concurrent access to sound snippets. Users may thus collaboratively explore the sound corpus represented by 3 audio descriptors directly mapped to the spatial dimensions, as displayed in Figure 3.

For the performers' gestures to be perceived as musically meaningful, we mapped the vertical dimension to frequency-related descriptors (such as the spectral centroid) and the z (depth) dimension to energy-related descriptors (such as the rms), which proved to be intuitively playable and easily understandable by the audience.

Although this interaction model may seem obvious, sparsity and uneven population of the descriptor space makes navigation through the units difficult, as already noted by [14, 15].

4.2 Interactive mosaicing

A second mapping strategy was implemented to allow the performers to explore target-driven audio mosaicing. It is necessary to incorporate a virtual time pointer in the interface, for example as a given path which the user should follow to retrieve the minimum distance audio units for a faithful reconstruction of the target. This pointer is then sent to CataRT as the desired target position, and CataRT itself then chooses the closest matching units from the corpus.

The spatial trajectory associated with this target should be preset in advance. If it is the user who chooses a suitable, continuous path, it could, for example, define a pictorial shape which may have a semantic relationship with the target. For example, when reconstructing a whispered voice from sea recordings, drawing a shape halfway between a lip and a wave turned out to be rather suggestive.

The new mapping is schematized in Figure 4. Note how the cloud along the predefined XY path (the target path) now consists of slices in YZ, normal to the path plane. For each target frame, the YZ plane displays all the corpus units located according to the descriptor distances.

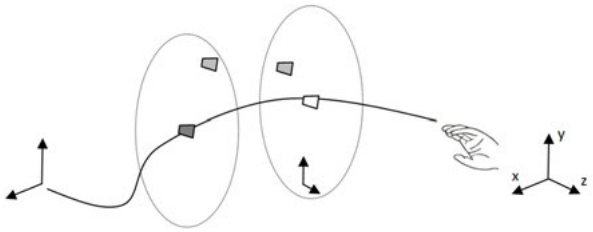


Figure 4: Gesture mapping for realtime target-based mosaicing

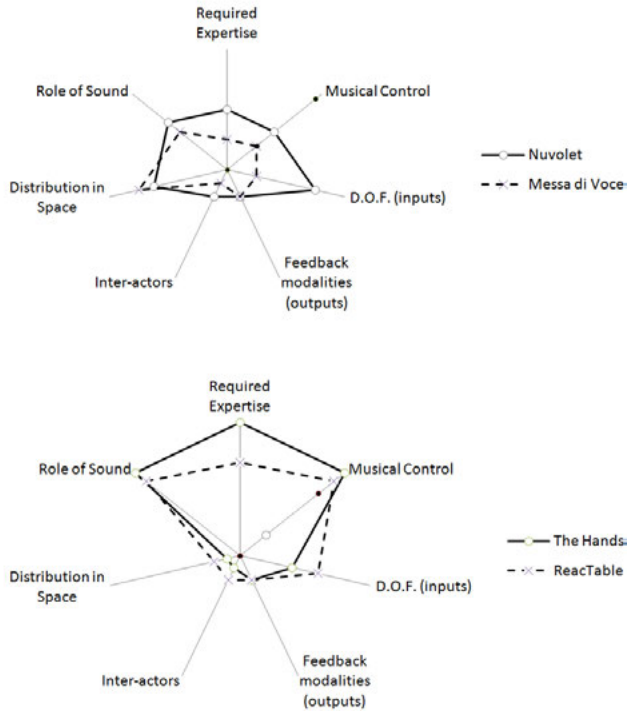


Figure 5: dimension space plots for four different gestural interfaces

This new mapping provided a richer experience to the performers, but concurrent use of the interface was less obvious.

5. DISCUSSION

The presented interface lies between the archetypes of a musical instrument and an interactive installation. Depending on the specific focus, its reference paradigm can range from sonification of gestural information to gesture-based sonic exploration. Figure 5 displays the 7-axis dimension space plots (see [1]) of Nuvolet, a virtuosistic interface like M. Waisvisz's The Hands, Jaap Blonk's solo from Messa di Voce, a performance and installation for voice and interactive visuals [7] and the collective tangible interface ReacTable, from Sergi Jordà. Nuvolet lies between The Hands and Messa di Voce, which also explores the realms of voice visualization, but is clearly shifted towards a sound installation profile.

The very nature of the interface imposes severe constraints on a number of desirable features for musical instruments, as listed in [12], like generality or perceivable correspondence between performer *effort* and sound quality. Moreover, the lack of the haptic channel for feedback in open-

air controllers increases the cognitive demands in such interfaces [17], which makes the trade-off between precision and agility even harder. These issues are most successfully addressed in smaller, instrumental interfaces, as with the Silent Drum Controller [11].

We observed however that through practice, performers were gradually less dependent on the visual cues and relied more on proprioception and kinesthetic feedback, but at the cost of being too static onstage.

6. CONCLUSIONS

We presented Nuvolet, a gestural interface for collaborative exploration of sound clouds and interactive target-based audio mosaicing. Nuvolet offers direct manipulation of multidimensional representations of sound corpuses by moving the hands on the space. A number of mapping strategies were studied, as well as an evaluation of the defining features of the new interface. Despite its rather satisfactory behavior in a restricted context, the challenges inherent in the design of open-air interfaces pose some unavoidable issues which deserve further research.

A logical addition to the interface could be a simultaneous gestural control of the sound spatialisation and an adoption of the general GDIF OSC namespace for exchanging of gesture related information between the software modules, in the line of [9], as well as the definition of a semantic mapping layer if more complex and multiuser input gestures are adopted [8] .

7. ACKNOWLEDGMENTS

The authors would like to thank all the members of the Esmuc Laptop Orchestra for their collaboration and continuous feedback throughout the development and testing of Nuvolet.

8. REFERENCES

[1] D. Birnbaum, R. Fiebrink, J. Malloch, and M. M. Wanderley. Towards a dimension space for musical devices. In *Proceedings of the New interfaces for Musical Expression (NIME)*, pages 192–195, 2005.

[2] W. Burt and A. Thompson. Fair exchanges'. *Writings on Dance V*, 1990.

[3] C. Casciato. On the choice onf gestural controllers for musical applications: An evaluation of the lightning ii and the radio baton. Master's thesis, McGill University, 2007.

[4] G. Coleman. Mused: Navigating the personal sample library. In *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, August 2007.

[5] C. Jacquemin, R. Ajaj, R. Cahen, Y. Olivier, and D. Schwarz. Plumage: design d'une interface 3d pour le parcours d'échantillons sonores granularisés. In *Proceedings of the 19th International Conference of the Association Francophone d'Interaction Homme-Machine*, pages 71–74. ACM, 2007.

[6] J. Janer, M. Haro, G. Roma, T. Fujishima, and N. Kojima. Sound object classification for symbolic audio mosaicing: A proof-of-concept. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 297–302, Porto, Portugal., July 2009.

[7] G. Levin and Z. Lieberman. In-situ speech visualization in real-time interactive installation and performance. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, pages 07–09, 2004.

[8] J. Malloch, S. Sinclair, and M. Wanderley. From controller to sound: Tools for collaborative development of digital musical instruments. In *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, August 2007.

[9] M. Marshall, N. Peters, A. Jensenius, J. Boissinot, M. Wanderley, and J. Braasch. On the development of a system for gesture control of spatialization. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 260–266, New Orleans, USA, August 2006.

[10] J. Mustard. *The integrated sound, space and movement environment: The uses of analogue and digital technologies to correlate topographical and gestural movement with sound*. PhD thesis, Western Australian Academy of Performing Arts, 2006.

[11] J. Oliver and M. Jenkins. The silent drum controller: A new percussive gestural interface. In *Proceedings of the International Computer Music Conference (ICMC)*, 2008.

[12] G. Paine. Towards unified design guidelines for new interfaces for musical expression. *Organised Sound*, 14(02):142–155, 2009.

[13] J. Paradiso. The brain opera technology: New instruments and gestural sensors for musical interaction and performance. *Journal of New Music Research*, 28(2):130–149, 1999.

[14] D. Schwarz. *Data-driven concatenative sound synthesis*. PhD thesis, Université Paris, 2004.

[15] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton. Real-time corpus-based concatenative synthesis with catart. In *Proceedings of Digital Audio Effects (DAFx)*, Montreal, Canada, September 2006.

[16] G. Vigliensoni. The enlightened hands: navigating through a bi-dimensional feature space using wide and open-air hand gestures. In *Proceedings of the New interfaces for Musical Expression (NIME)*, Sidney, Australia, June 2010.

[17] G. Vigliensoni and M. Wanderley. Soundcatcher: Explorations in audio-looping and time-freezing using an open-air gestural controller. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 100–103, New York, USA, 2010.

[18] T. Winkler. Making motion musical: Gesture mapping strategies for interactive computer music. In *Proceedings of the International Computer music Conference (ICMC)*, pages 261–264, Banff, Canada, 1995.

[19] T. Winkler. Motion-sensing music: Artistic and technical challenges in two works for dance. In *Proceedings of the International Computer Music Conference (ICMC)*, Ann Arbor, USA, 1998.

[20] A. Zils and F. Pachet. Musical mosaicing. In *Proceedings of Digital Audio Effects (DAFx)*, Limerick, Ireland, December 2001.

Effective and expressive movements in a French-Canadian fiddler's performance

Erwin Schoonderwaldt
Inst. of Music Physiology and Musicians' Medicine
Univ. of Music, Drama and Media Hanover
Emmichplatz 1
D-30175 Hannover, Germany
erwin.schoonderwaldt@hmtm-hannover.de

Alexander Refsum Jensenius
fourMs, Dept. of Musicology
University of Oslo
P.O. Box 1017 Blindern
NO-0315 Oslo, Norway
a.r.jensenius@imv.uio.no

ABSTRACT

We report on a performance study of a French-Canadian fiddler. The fiddling tradition forms an interesting contrast to classical violin performance in several ways. Distinguishing features include special elements in the bowing technique and the presence of an accompanying foot clogging pattern. These two characteristics are described, visualized and analyzed using video and motion capture recordings as source material.

Keywords

fiddler, violin, French-Canadian, bowing, feet, clogging, motion capture, video, motiongram, kinematics, sonification

1. INTRODUCTION

Recent developments of motion capture technologies have spurred the interest in movement analysis of music performances. Measurement of instrumentalists' movements have been used for the study of effective movements related to the production of sound [2, 3, 5], as well as the study of ancillary movements related to expression [2, 12]. Motion capture measurements have proven particularly useful for the study of bowed-string instrument performance, as the sound in these instruments is entirely produced by means of overt movements [1, 4, 7, 10, 14].

The violin is known as a highly versatile instrument being played in a wide range of musical styles and traditions. Yet, there is a strong bias of performance research focusing on classically trained, expert performers. Despite the large variety of individual strategies among classical performers, this might constrain our scope on the wide range of possibilities offered by the instrument in terms of playing technique and expressivity.

In this paper we present a case study of a fiddler's performance, stemming from the French-Canadian tradition. The fiddling tradition largely contrasts the classical: a) Performances mostly take place in informal settings, for example a jam session in a pub. b) The playing technique is typically less "polished," as the sound should be audible under varied acoustic conditions, including noisy environments, outside, etc. c) The repertoire mainly consists of traditional tunes learned by ear. d) The performances have an improvised character and are often combined with dance. e) Fiddlers often accompany their playing with clogging patterns produced with their feet, in the French-Canadian tradition typically a heel-heel-toe-toe pattern in sixteenth notes.

The aim of this paper is twofold. First, we want to illustrate the violin's versatility as a musical instrument by focusing on a non-classical violin performance. Second, we want to provide a showcase for alternative analysis methods and representations of movement data. We will present motion-based analyses of a variety of aspects of the performance, illuminating particular bowing techniques and clogging patterns. Motiongrams extracted from the video will be used to illustrate global features of the performance, while analysis of motion capture data will provide insights at a higher level of detail.

2. RECORDINGS

Performances of an expert fiddler of the French-Canadian tradition were recorded.[1] The fiddler is an experienced performer, playing and recording on a regular basis, and he is also active as a teacher. For the analyses in this paper a complete performance was selected of the reel *Le bedeau de l'enfer* (transl. *The deacon from hell*). The reel consists of two parts (A and B), and was performed with the following repetition scheme in the recording: A-A-B-B-A-A-B-B. The average tempo of the performance was ∼103 BPM.

Motion capture recordings were made at IDMIL/McGill using a Vicon 460 system with six cameras placed around the performer. Video and audio were recorded in synchrony with the motion capture data. Full body measurements were made using the Plugin-gait marker placement (39 markers). The position and orientation of the violin and the bow were tracked by five markers on each object. Additional sensors were placed on the bow for measuring bow force and acceleration [4]. This setup allowed for an accurate calculation of all relevant bowing parameters, including bow velocity, bow-bridge distance and bow force, as well as the angles of the bow relative to the violin [11].

3. GLOBAL FEATURES

As a first step in the analysis, global features were studied using the video recording as source material.

3.1 Video analysis

Motiongrams created from the video recordings of the fiddle performance are shown in Fig. 1. The motiongram technique is based on calculating the normalized mean value of

[1] See recordings of Fiddler performances at http://www.youtube.com/schoondw

NIME'11, 30 May–1 June 2011, Oslo, Norway.

each column or row in a *motion image* (frame-difference image) [6]. As opposed to traditional keyframe displays, where individual video frames are plotted next to each other, motiongrams can show the temporal and spatial unfolding of movement over time.

The horizontal motiongram in Fig. 1, which shows information about movement in the vertical plane, provides the clearest information of the rhythmical body movements during the performance. The upper part mainly shows the movement of the bowing arm, whereas the lower part represents the clogging pattern produced by the legs. Interestingly, a clear transition in the clogging pattern can be distinguished halfway through the performance, which coincides with the reprise of part A of the reel. The transition marks an increased intensity of the performance.

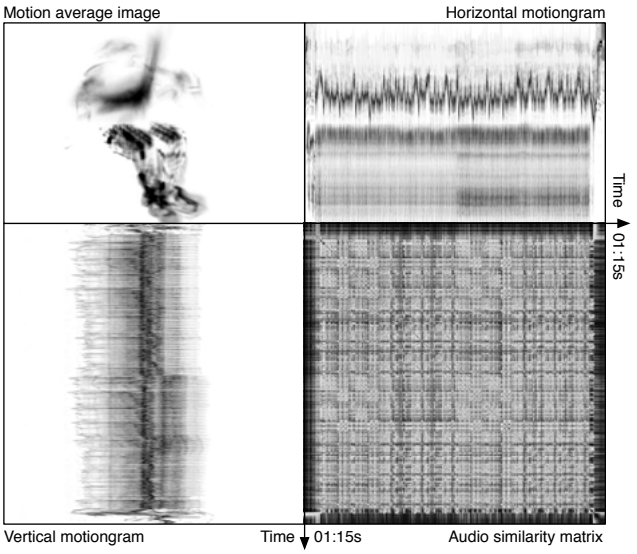


Figure 1: Motion average image, horizontal and vertical motiongrams of the video recording, and a similarity matrix of the spectrogram of the audio recording.

The *motion average image* in Fig. 1 is based on calculating the mean matrix of all motion image frames. The result is a "blurred" image displaying the spatial distribution of motion for the whole recording. Here we can see the movement of the bowing arm and the pronounced movement of the legs corresponding to the clogging. Furthermore, it can be seen that the range of motion in the right leg was larger than that in the left leg.

As a reference to sound, we have also included a similarity matrix of the sound recording in Fig. 1. Since time runs in two dimensions in a similarity matrix, it can be used to compare sonic features to motion features in both motiongram directions. Notice how the repetition structure of the performance (A-A-B-B-A-A-B-B) can be clearly distinguished in the similarity matrix.

3.2 Sonification of motiongrams

Since motiongrams share many visual properties with spectrograms, they can be used as the basis for an 'inverse spectrogram' approach, as suggested in [13]. This way we can create a direct mapping from motiongram to spectrogram.

An implementation of such an inverse spectrogram technique for sonification of motiongrams is based on reading each row in the motiongram matrix and mapping them directly to an interpolated oscillator bank, which does the additive synthesis. The final result is a direct sonification of the motion, where low frequencies are controlled by movements in the lower part of the image and higher frequencies by movements in the upper part.

An example of a sonification based on the video recording of the fiddler shows how present both the bowing and clogging patterns are in the sonification.[2]

4. MOTION CAPTURE ANALYSIS

A selection of particular features of the performance were analyzed in more detail using the 3D motion capture data.

4.1 Bowing

The extracted bowing parameters allowed for some general observations regarding the use of the bow. The piece mainly consists of 16th notes, which were played détaché or in short two- or three-note slurs. The range of bowing parameters was mainly in a "comfortable zone;" the middle of the bow was used, the bow velocity was ∼1 m/s, and the bow-bridge distance was ∼4 cm. A rather large range of bow force was used, with pronounced peaks revealing a strong accentuation pattern.

A peculiar feature of the bowing was that the performer used a pronounced backward tilt (i.e. *rotation around the length axis*) of the bow. Interestingly, a backward tilt of the bow is uncommon and even discouraged in classical playing, as it creates a rough sound. An acoustical explanation for this is related to partial slips, causing spikes in the bridge force signal making up the sound [8, 9]. The consistent use of backward tilt suggests that this roughness might be a desirable aspect of the sound quality in fiddling.

Another interesting element present in the performance was the use of the *shake*, a type of bowing ornament commonly used in French-Canadian fiddling as well as other fiddling styles. The shake could be characterized as "a rapid and indistinct bowed triplet, more of a scratch than a series of notes."[3]

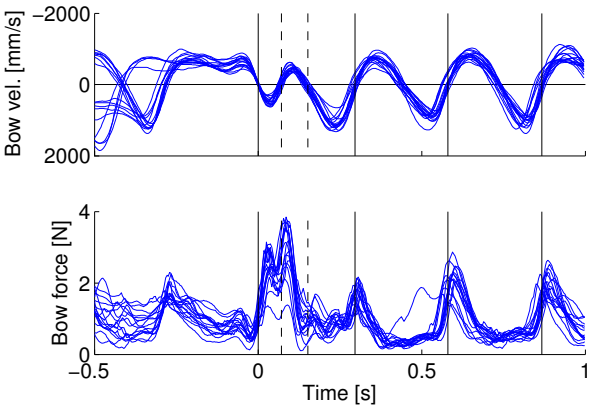


Figure 2: (a, b) Overlapping bow velocity and bow force profiles of 16 occurrences of the main motif, consisting of a shake (at t=0) followed by four 16th notes. The solid vertical lines indicate half beat durations (8th note level). The dashed vertical lines show the subdivision of the shake.

Fig. 2 shows overlapping bow velocity and bow force profiles of 16 selected occurrences of the main motif of the piece consisting of a shake followed by four 16th notes. The timing profiles extracted from the bow reversals (zero crossings in bow velocity signal) are shown in Fig. 3. The profiles

[2]Video at http://www.youtube.com/watch?v=pV8JglqB94k
[3]Definition found on an internet discussion forum.

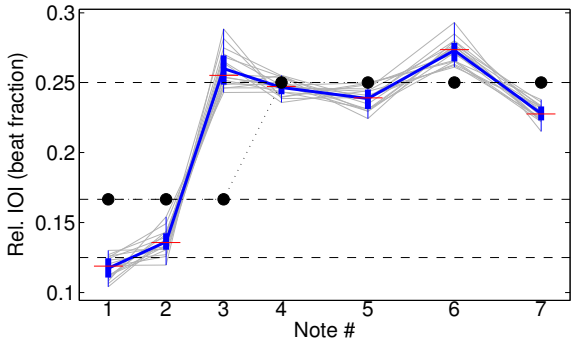


Figure 3: Timing profiles of the main motif shown in Fig. 2. The dashed lines indicate the nominal values of 8th notes, triplets, and 16th notes. Nominal durations of the notes (according to a transcription) are indicated by black dots.

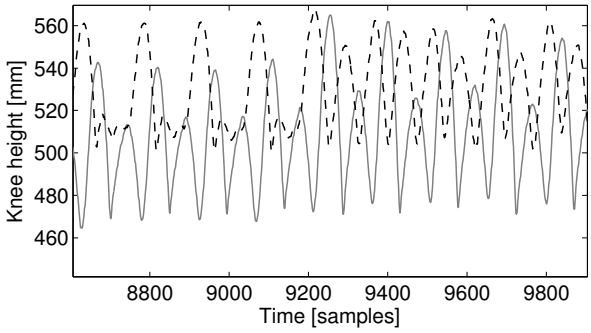


Figure 4: Height of right knee (grey) and left knee (dashed) versus time. The selected intervals show the different clogging patterns present in the performance, as well as the transition between them.

reveal a high degree of consistency, both with regard to the use of the bowing parameters, as well as the rhythmic performance. The shake was played as two rapid notes (32nd) followed by an 8th note. The two first notes were played with a high bow force (2–4 N), showing a pronounced double peak; in combination with the small bow displacement this resulted, indeed, in a scratchy sound with a more or less percussive character.

Figs. 2 and 3 also give insight in the accentuation pattern of the 16th notes. The first and the third notes (played up bow) were accented; the strongest accent fell on the third note, resulting in a syncope-like effect. The accentuation was achieved by a combination of (1) bow force, showing strong peaks at the attack of the first and third notes; (2) asymmetry in the bow velocity pattern, showing a shorter attack time (steeper slope) at the beginning of the first and the third notes; and (3) prolongation of the first and the third note at the cost of notes two and four; the third note was consistently played the longest, whereas the fourth note was consistently played the shortest.

4.2 Clogging

The perhaps most special feature of the fiddling performance presented here is the clogging with the feet, used as a rhythmic accompaniment of the playing. As already suggested by the horizontal motiongram in Fig. 1, there was a transition in the clogging pattern in the middle of the performance. The motion capture data reveals that in the first part the beats were divided in a long-short-short rhythm, with tapping pattern: right heel / right toe / left foot. In the second part the beats were subdivided in four 16th notes with tapping pattern: right heel / left heel / right toe / left toe.

The patterns, as well as the transition between them, are illustrated in Fig. 4 by the vertical movement of the knees. The right leg shows the largest movement amplitude, confirming the initial observation from the motion average image in Fig. 1. An explanation for this is that the player stressed the strong metrical events (first and third within a group of 16th notes), which coincide with the tapping of the right leg. In the first part, the left foot taps on every fourth within a group of 16th notes. In the second part the movement pattern of the left leg is similar to that of the right leg, but shifted in phase by one 16th note so that the taps are alternating. The transition seems to occur suddenly from one stable coordination pattern into another, which indicates a highly efficient motor control.

The kinematic profile of the right leg (knee, heel, toe; seen from the side) is shown in Fig. 5. The position of the leg is marked in bold at instances of tapping with heel and toe. The motion consists of a combined up-down and forward-backward movement of the lower leg: the leg moves forward before tapping with the heel, and backward before tapping with the toe. An interesting detail here is that in preparation for the heel tap the leg is pushed up by flexion of the foot after the toe tap, minimizing the effort of lifting the leg.

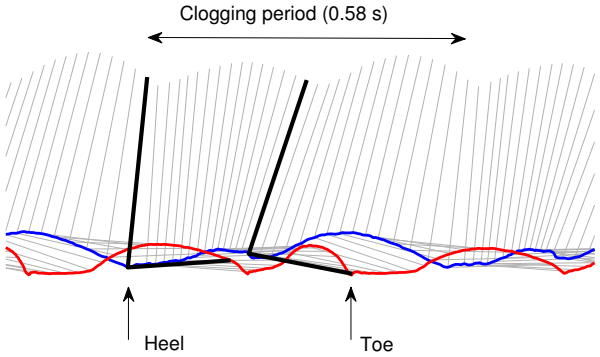


Figure 5: Kinematic representation of the clogging pattern of the lower right leg. The sticks show the connections between knee, heel and toe markers. Movement trajectories are shown for the heel and the toe. The tapping events of the heel and the toe are marked by arrows. The time interval between the shown frames is 20 ms between frames; a translation to the left with increasing time is applied for a clear presentation of how the movement unfolds.

The moments of impact of the feet were extracted from the vertical displacement of the heel and the toe markers of the right and the left foot, using a peak picking algorithm (with knowledge of the periodicity of the signals). This allowed for extracting timing characteristics of the clogging patterns, shown in Fig. 6. The clogging pattern in the first part shows a clear long-short-short pattern, consisting of an 8th note interval followed by two 16th note intervals. The first of the two short intervals was consistently prolonged at the cost of the second one. The clogging pattern in the second part showed rather regular 16th note intervals. The first

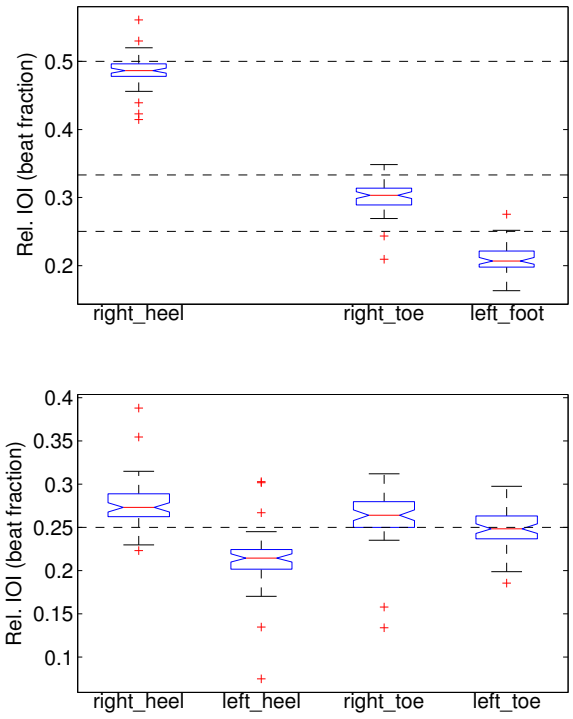


Figure 6: Timing profiles (normalized IOI) of the two clogging patterns. The dashed lines indicate nominal values of 8th notes (0.5) and 16th notes (0.25).

and the third intervals were slightly prolonged, consistent with the timing profile of the bowing in Fig. 3. However, in the clogging the first interval was longest, and not the third as in the bowing.

5. DISCUSSION AND CONCLUSIONS

In this study video recordings and 3D motion capture data have been used to reveal different aspects of a fiddler's performance. Motiongrams and a motion average image, in combination with a sound similarity matrix, were used to distinguish global patterns in the performance. They revealed the spatial distribution of motion in the performance, as well as a transition in the clogging pattern halfway through the performance. Analysis of the motion capture data provided detailed information of special performance features, including the shake, and kinematic features of the clogging and timing profiles.

Analysis of the bowing provided some indications of differences between fiddling and classical performance, which might be traced back to a different sound ideal. Three distinct features regarding the use of the bow we described are: (1) the backward tilting of the bow, (2) the accentuated use of bow force in détaché bowing, and (3) the "shake," a special ornamentation technique. These features contribute to a highly articulated sound containing scratchy elements, whereas classical performers generally strive for a more homogeneous and "polished" sound quality. Even though the data of only one player have been shown, such features can be commonly observed in other fiddlers' performances within the same and other traditions (e.g. Celtic, Scandinavian).

Besides the sound-producing effects, the performer's movements may also be interpreted at an expressive level, providing the performance its special character and style. The bow velocity and bow force profiles provide insight in the expressive shaping of the performance by means of accentuation and expressive timing. The combination of clogging and bowing also provides a nice illustration of a complex coordinated motion involving all four limbs.

While the violin is not a new instrument for musical expression, its use in fiddling and in combination with other musical elements (e.g. clogging) provides for a different musical experience. We hope that alternative performance studies of other (traditional) instruments may open for a broader understanding of musical instruments and their use in performance.

Acknowledgments

Thanks to Pascal Gemme for the fiddling, and Marcelo Wanderley for making the recordings possible. The study was partly financed by NSERC-SRO. Currently, the first author receives an Alexander von Humboldt postdoctoral fellowship, and the second author is supported by a grant from the Norwegian Research Council.

6. REFERENCES

[1] A. P. Baader, O. Kazennikov, and M. Wiesendanger. Coordination of bowing and fingering in violin playing. *Cognitive Brain Research*, 23(2-3):436–443, 2005.

[2] S. Dahl. *On the beat: Human movement and timing in the production and perception of music*. PhD thesis, KTH – School of Computer Science and Communication, Stockholm, Sweden, 2005.

[3] S. Dahl and E. Altenmüller. Motor control in drumming: Influence of movement pattern on contact force and sound characteristics. In *Proceedings of Acoustics '08*, Paris, France, 2008.

[4] M. Demoucron. *On the control of virtual violins: Physical modelling and control of bowed string instruments*. PhD thesis, Université Pierre et Marie Curie (UPMC), Paris & Royal Institute of Technology (KTH), Stockholm, 2008.

[5] W. Goebl and C. Palmer. Anticipatory motion in piano performance. *J. Acoust. Soc. Am.*, 120(5):3004–3004, 2006.

[6] A. R. Jensenius. Using motiongrams in the study of musical gestures. In *Proceedings of the 2006 International Computer Music Conference*, pages 499–502, New Orleans, LA, 2006.

[7] E. Maestre. *Modeling Instrumental Gestures: An Analysis/Synthesis Framework for Violin Bowing*. PhD thesis, University Pompeu Fabre, Barcelona, Spain, 2009.

[8] M. E. McIntyre, R. T. Schumacher, and J. Woodhouse. Aperiodicity in bowed-string motion. *Acustica*, 49:13–32, 1981.

[9] M. E. McIntyre, R. T. Schumacher, and J. Woodhouse. Aperiodicity in bowed-string motion: on the differential-slipping mechanism. *Acustica*, 50:294–295, 1982.

[10] E. Schoonderwaldt. *Mechanics and acoustics of violin bowing: Freedom, constraints and control in performance*. PhD thesis, KTH – School of Computer Science and Communication, Stockholm, Sweden, 2009.

[11] E. Schoonderwaldt and M. Demoucron. Extraction of bowing parameters from violin performance combining motion capture and sensors. *J. Acoust. Soc. Am.*, 126(5):2695–2708, 2009.

[12] M. M. Wanderley, B. W. Vines, N. Middleton, C. McKay, and W. Hatch. The musical significance of clarinetists' ancillary gestures: An exploration of the field. *Journal of New Music Research*, 34(1):97–113, 2005.

[13] W. S. Yeo and J. Berger. Application of image sonification methods to music. In *Proceedings of the International Computer Music Conference*, Barcelona, 2005.

[14] D. Young. *A methodology for investigation of bowed string performance through measurement of violin bowing technique*. PhD thesis, Massachusetts Institute of Technology, 2007.

Flowspace – A Hybrid Ecosystem

Daniel Bisig
Zurich University of the Arts
Institute for Computer Music and Sound Technology
Baslerstrasse 30
8048 Zurich, Switzerland
daniel.bisig@zhdk.ch

Jan Schacher
Zurich University of the Arts
Institute for Computer Music and Sound Technology
Baslerstrasse 30
8048 Zurich, Switzerland
jan.schacher@zhdk.ch

Martin Neukom
Zurich University of the Arts
Institute for Computer Music and Sound Technology
Baslerstrasse 30
8048 Zurich, Switzerland
martin.neukom@zhdk.ch

ABSTRACT

In this paper an audio-visual installation is discussed, which combines interactive, immersive and generative elements. After introducing some of the challenges in the field of Generative Art and placing the work within its research context, conceptual reflections are made about the spatial, behavioural, perceptual and social issues that are raised within the entire installation. A discussion about the artistic content follows, focussing on the scenography and on working with flocking algorithms in general, before addressing three specific pieces realised for the exhibition. Next the technical implementation for both hard- and software are detailed before the idea of a hybrid ecosystem gets discussed and further developments outlined.

Keywords

Generative Art, Interactive Environment, Immersive Installation, Swarm Simulation, Hybrid Ecosystem

1. INTRODUCTION

This publication describes an installative artwork entitled "Flowspace" that was realised by the authors and shown to the public in the context of a thematic exhibition about sound, space and virtuality [7]. The installation creates an interactive, immersive, and generative environment for audiovisual compositions that are controlled via simulations of swarm behaviour. As such, the installation situates itself within the fields of Generative Art and Artificial Life. One of the most fundamental challenges in Generative Art relates to the establishment of meaningful and traceable mapping relationships between the underlying algorithmic processes and the resulting aesthetic output [2]. "Flowspace" shifts the focus away from the mapping issue in favour of an approach that places a stronger emphasis on the customization of the generative algorithms themselves in order to match a particular artistic goal [10]. The issue of interaction with complex autonomous systems constitutes another fundamental challenge in Generative Art. “Flowspace” approaches this challenge by providing an interaction model that is based on multiple levels of immediacy in control and feedback. "Flowspace" employs generative algorithms not only for the creation of aesthetic feedback but also to establish coherence among spatial, perceptual, behavioural and social phenomena that manifest themselves within the installation. We employ the term hybrid ecosystem to describe the characteristics of such an installative environment. This designation is related to the term hybrid ecology as it has been coined by Crabtree and Rodden [5], since both of them refer to the creation of collaborative situations in mixed reality environments. Rising interest in ecological approaches to musical composition [12,6] and recent examples in installation art [1,11] are a strong indication that this approach indeed constitutes a promising direction for Interactive Media and Generative Art.

2. BACKGROUND

The installative artwork "Flowspace" represents a tangible result from two consecutive research projects that are conducted at the Institute for Computer Music and Sound Technology of the Zurich University of the Arts. The first project is entitled ISO – Interactive Swarm Orchestra – and its successor project is entitled ISS – Interactive Swarm Space. Both projects explore strategies for interrelating swarm simulations with the interactive and aesthetic properties of an artwork [2,4]. Furthermore, the projects try to promote artistic applications of swarm simulations by developing open-source tools in software and hardware that aid in the realisation of swarm based artworks [3,13].

3. CONCEPT

The realisation of “Flowspace” reflects our intention to create a hybrid environment in which the natural and simulated properties and behaviours of the space and its inhabitants overlap and interrelate. This situation creates an immersive experience that involves spatial, behavioural, perceptual and social aspects, which are described in more detail in the following sections.

3.1. Spatial Aspects

The architectural structure of “Flowspace” is realised in the shape of a Dodecahedron [see figure 1]. The shape of the installation conforms to the characteristics of the installation's generative feedback [see section 5.1.]. As a result, the architecture of the installation supports the blending of physical and virtual space. The simulation space overlaps with the installation space that surrounds the visitors. In addition, the simulation space is mapped onto a two-dimensional segment of the Dodecahedron surface and forms part of the installation's interface. This enables the visitors to experience a spatial immersion within the virtual swarm and to simultaneously assume a juxtaposed position outside of the swarm.

3.2. Behavioural Aspects

In “Flowspace”, the behaviours of visitors and the swarm agents affect each other on multiple levels that differ in immediacy and spatial extension. By touching the surface of the interface, visitors can directly manipulate the positions of particular agents. Other agents subsequently respond to these changes. These interrelating agent behaviours transform the visitors' interactions from an initially local and immediate

effect into an element in the emergent dynamics of the installation's audiovisual compositions. Different combinations of properties exist for swarm simulation and audiovisual processes and are organized as discrete states in a finite state machine. The selection of the states is controlled by the visitors' long term accumulated activities. The installation's characteristics as a hybrid ecosystem results from the interrelations among the activities of its natural and virtual inhabitants that occur on several temporal, spatial and causal levels. The simplicity and immediacy of the interface's physical manipulation and its subsequent effect on the installation's responses provides a natural form of interaction, which helps to balance the visitors' intuition, familiarity, curiosity and surprise.

3.3. Perceptual Aspects

“Flowspace” provides feedback through the modalities of touch, hearing, and vision. Correlations among these modalities shape the aesthetic experience, direct the visitors' attention and influence the traceability of the installation's behaviours. In “Flowspace”, the audiovisual compositions and the visual and tactile feedback of the interface are all linked via the swarm simulation. Again, multiple levels of immediacy exist in the creation of the installation's output. In order of decreasing immediacy, they range from the very basic tactile experience from touching the interface, the presence of bright circles underneath the visitors' fingers, the abstract graphical depictions of the swarm simulation on the interface, to the presentation of the audiovisual compositions themselves [see figures 2-4]. In addition, these perceptual phenomena also differ with respect to their spatial characteristics. The most immediate feedback of the visitors' hand movements is localized on the surface of the interface. The presentation of the audiovisual compositions is spatially distributed and forms part of the visitors' immersive experience.

3.4. Social Aspects

In “Flowspace”, the installation space and its interface are sufficiently large to allow several people to become involved at the same time. Due to the installation's relatively open forms of interaction and exploration, different social situations may appear. Some social settings resemble performance situations when individual visitors become performers that actively interact with the interface while the remaining visitors act as an audience. Other social settings are more collaborative in that most of the visitors try to collectively affect the installation's behaviour. The fact that various social situations appear and disappear forms part of the installation's characteristics as a hybrid environment.

4. ART WORKS

The “Flowspace” installation was part of an exhibition entitled “Milieux Sonores” that was shown in two separate occasions: in the Kunstraum Walcheturm in Zürich in 2009 and in the Gray Area Foundations for the Arts in San Francisco in 2010.

4.1. Scenography

The scenographical integration of the installation into the environment of the exhibition was realized in close collaboration with the curator Marcus Maeder. The crystal-like characteristics of the Dodecahedron shape [see figure 1] is partially resumed in the form of black wooden shards that gradually transform the exhibition space into the spatial situation of the installation.

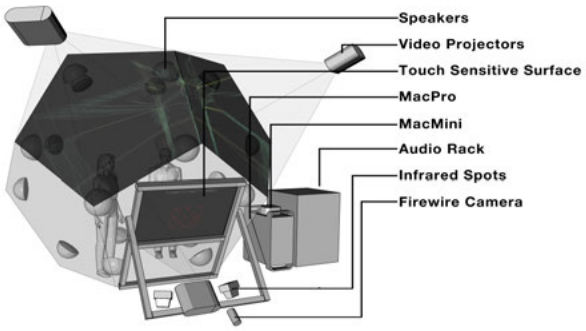


Figure 1: Schematic Representation of the Installation

4.2. Swarm-based Artworks

The installation forms an environment for the interaction with three different swarm-based artworks. The swarm simulations are displayed as simple graphical renderings on the surface of the interface. Visual compositions are projected as panoramic imagery on two pentagonal surfaces above the interface. Musical compositions are spatialised via speakers that completely surround the visitors. The detailed implementation of the swarm simulations and the audio generation mechanisms is described in a different publication [10].

4.2.1. Impacts

Starting from a strongly interactive premise, the flocking algorithm in this piece explores the possibility of hierarchical relationships between several flocks, similar to the interdependence within an ecosystem or food chain. There are three types of entities present in the "Impacts" model: the first type of agent is the attractor. Its behaviour is fully dependent on the visitor’s action since it can't move by itself but is displaced by the visitor's touch. The agents of the secondary swarm influence their own kind and react to the attraction forces of the first swarm. They serve as attractors to the agents within the third swarm. The behaviours of the agents within the second and third swarms are parameterized in such a manner as to create very dynamic motion patterns. The music consists of a background layer of Ambisonic ambience of the Notre Dame cathedral in Paris. The individual collisions between agents in the swarm simulation trigger piano samples on impact and a granular echo of the same pitch when an escape point is reached. In the true spirit of emergent structures, the mixture of all of these events alone is what generates the characteristic texture of sounds. The user interaction controls the choice of pitches: the higher the level of interaction, the more active the

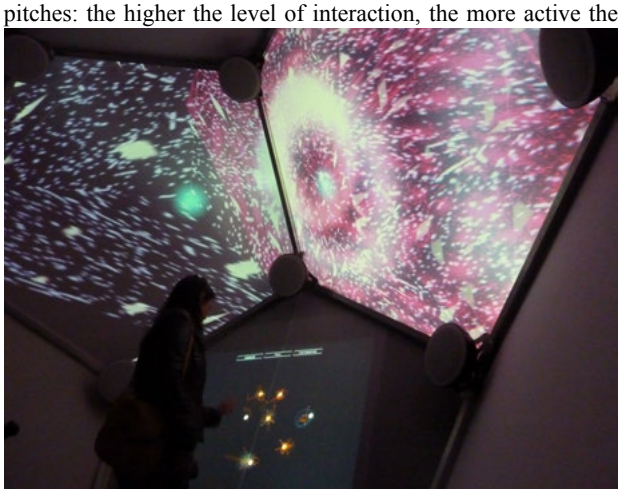


Figure 2: The “Impacts” Piece

entire swarm simulation is and the richer and more dissonant the pitch-sets become. Since the note-events are spatialised according to the position of the generating agents, these pitch clusters are perceivable as being located in certain sectors of the surround field. The visualisation re-interprets the idea of impacts and escape points by tracing these points into a Delaunay triangulation and showing growing concentric circles around the points of impact [see figure 2].

4.2.2. Flow

The "Flow" piece exploits the existence of periodically recurring events within the simulation in order to generate rhythmical structures in the acoustic and visual output. These recurring events originate from repeated changes in the neighbourhood relationships between two different swarms: a predominantly static swarm whose agents are attached to the visitors' touch positions and a highly dynamic swarm, that traverses the static swarm. As long as the visitors do not move the static agents, the dynamic agents settle into cyclic trajectories that cause them to periodically approach the static agents and thereby trigger the generation of sound grains whose content is created via additive synthesis. The duration and acoustic spatialisation of the grains and the frequencies of the oscillators is controlled by the dynamic agents' position, velocity and jerk. Accordingly, the musical motif is dominated by stable rhythmic patterns whereas the texture of the individual sounds constantly varies. The visual output renders the static agents as a mesh of lines that interconnect the agents' positions [see figure 3]. The discrepancy between the large scale repetitions and local variations in the trajectories of the dynamic agents is visually emphasized by drawing the trajectories as thin lines that rapidly widen into series of spokes according to the agents' jerk.

Figure 3: The "Flow" Piece

4.2.3. Membranes

The "Membranes" piece employs models of physical springs for both swarm simulation and sound synthesis in order to create a perceptual and aesthetic proximity between the two. The simulation consists of two types of swarms: a static swarm that is directly manipulated by the visitors, and a dynamic swarm whose agents behave as end points of interconnected springs. Depending on the distance between spring agents, new springs are created or old springs are destroyed. The static agents repel the spring agents. Whenever the visitors move the static agents, the previously established network of interconnected springs is disrupted. The musical algorithm employs a non-linear model of a physical spring for sound synthesis [8]. Each of these acoustic springs corresponds to a spring in the swarm simulation. The movement of the spring agents drives the excitement of the acoustic springs. Whenever an agent spring is created or destroyed, a strong excitement is applied to the acoustic spring. The location of the acoustic spring in the sound-field is determined by the centre position of the spring agent. The musical output consists of a slowly undulating texture that is occasionally interrupted by discrete and loud sounds that result from the creation and destruction of springs. The visual output displays the connectivity of the springs' mesh as stacks of triangles and the small fluctuations of the springs' mass points as interconnected lines the follow the points' trajectories [see figure 4].

Figure 4: The "Membranes" Piece

5. IMPLEMENTATION

The implementation of the "Flowspace" installation relies on hard- and software tools that have been developed in the context of the ISO/ISS research projects.

5.1. Hardware Setup

The structure of the installation [see figure 1] is built from an aluminium frame that is about 4.2 meter in diameter and has the shape of a Dodecahedron. This shape was initially chosen because of its suitability for positioning loudspeakers in a spherical arrangement for three-dimensional ambisonic sound projection [9]. Later on, the frame was extended for video rear projection by covering its surface with projection screens. The latest improvement consists of the integration of a tactile surface into one of the Dodecahedron's pentagonal faces. The video projection setup consists of three ultra-short throw projectors. The projection surface covers three neighbouring pentagonal surfaces. The two upper surfaces are used for a panoramic video projection of the visual compositions and the lower surface is used for the interface display. The touch interface is based on video tracking with rear diffuse infrared illumination.

5.2. Software Setup

The software part of the installation consists of a number of applications for swarm simulation, finger tracking, audio and video generation and installation state control. Many of the applications rely on a set of open source C++ libraries that were developed as part of the ISO project. These so-called "ISO" libraries [3] are available for both Mac OS X and Linux operating systems and can be downloaded from the project website [13]. The swarm simulations for the three different audiovisual compositions are implemented with the "ISO Flock" library. Several audio applications generate the acoustic output of the installation. The analysis of the swarm data and the control of sound generation and spatialisation is implemented differently by the three artworks. The audio for

"Impacts" is created in Max/MSP whereas "Flow" and "Membranes" employ sound synthesis algorithms implemented with the "ISO Synth" library. Two of the three applications for the visual rendering of the swarm simulations are implemented using the "ISO Visual" library. The visualisation for “Impacts” as well as the finger tracking and the master state control software are implemented in openFrameworks [14]. The master state control software is in charge of managing the different installation states and acts as a communications hub between the simulations, tracking software and audio and video engines. Inter-application communication is based on the OpenSoundControl protocol.

6. RESULTS AND DISCUSSION

The installation "Flowspace" creates an environment in which natural and artificial entities and their respective physical and virtual surroundings merge into a hybrid ecosystem. Based on the positive feedback that we have received from visitors during the exhibition of the installation, we believe that this approach is successful in creating an engaging experience for the visitors. We attribute the installation's positive reception to several aspects that are inherently part of our notion of a hybrid ecosystem. Firstly, the installation provides an environment that encourages intuitive and explorative forms of interaction. We have emphasized this aspect by allowing the visitors to engage with the installation and to experience its reactions via several levels of immediacy and across different modalities. Secondly, the installation's spatial, behavioural and perceptual properties are correlated via a single underlying swarm model and thereby allow the visitors to experience the installation as a coherent whole. Thirdly, the ecosystem characteristics of the installation offers the visitors the ability to become involved on perceptual, behavioral and social levels. When this involvement is sufficiently intense, each of these levels achieves immersive qualities.

7. CONCLUSIONS AND OUTLOOK

We believe that the notion of a hybrid ecosystem can inform artistic approaches in creating interactive and immersive environments. The realization of such an environment is an inherently interdisciplinary endeavour that combines knowhow and methods from various fields such as Artificial Live, Generative Art, Interaction Design, and Scenography. Since ecological approaches in Generative Art are relatively new, a vast range of scientific questions and artistic challenges exists that should be addressed. In particular, we would like to explore the following aspects that up to now have played only a marginal role in our work: In “Flowspace”, the capabilities of the swarm simulations and the characteristics of the audiovisual feedback mechanisms are predefined and never change during an exhibition. Because of this, the short-term behaviour of “Flowspace” is surprising and engaging for visitors, but its long-term behaviour tends to be repetitive and predictable. It would be interesting to augment the installation with the capability to undergo long-term changes through learning or evolution. The aesthetics of the audiovisual compositions in “Flowspace” are largely defined by its authors. Visitors can explore these compositions within relatively narrow aesthetic boundaries. It could provide additional interest if the role of the visitor's creative contribution is strengthened by expanding the range of interaction-based effects both with respect to the compositions and the underlying simulations. Finally and most importantly, we believe that the hybrid ecosystem approach provides an excellent context to experiment with rarely used modalities and unconventional interfaces. In “Flowspace”, the usage of sonic, visual and tactile feedback and its combination with a touch sensitive surface is interesting mainly due their correlation via a common generative mechanism. Other than that, neither the interface nor the feedback modalities are very unconventional. We are currently in the process of designing different types of interfaces that are specifically adapted to interaction with a spatially distributed and highly dynamic entity such as a simulated swarm. These new interfaces will play a dual role as control interface and display of swarm activities and will employ the same modalities for input and output in order to bridge the gap between the physical and virtual aspects of the hybrid ecosystem.

8. REFERENCES

[1] Bartlem, E. Immersive Artificial Life (A-Life) Art. *Journal of Australian Studies.* 84, Perth, API Network, 2005.

[2] Bisig, D., and Neukom, M. Swarm Based Computer Music - Towards a Repertory of Strategies. In *Proceedings of the Generative Art Conference*. Milano, Italy, 2008.

[3] Bisig, D., Neukom, M., and Flury, J. Interactive Swarm Orchestra - A Generic Programming Environment for Swarm Based Computer Music. In *Proceedings of the International Computer Music Conference*. Belfast, Ireland, 2008.

[4] Bisig, D., and Unemi, T. Swarms on Stage - Swarm Simulations for Dance Performance. In *Proceedings of the Generative Art Conference*. Milano, Italy, 2009.

[5] Crabtree, A., and Rodden, T. Hybrid ecologies: understanding cooperative interaction in emerging physical-digital environments. *Personal and Ubiquitous Computing.* 12, 7, Springer, London, England, 2008.

[6] Davis, T. Cross-Pollination: Towards an aesthetics of the real. In *Proceedings of International Computer Music Conference.* SARC, Belfast, Ireland, 2008.

[7] Maeder, M. (Ed.). *Milieux Sonores - Klangliche Milieus. Klang, Raum und Virtualität.* Transcript Verlag, Bielefeld, Germany, 2010.

[8] Neukom, M. *Signale, Systeme und Klangsynthese - Grundlagen der Computermusik.* Peter Lang, Bern, Switzerland, 2005, 515 – 519.

[9] Schacher, J.C. and Kocher, P. Ambisonics Spatialization Tools for Max/MSP. In *Proceedings of the International Conference on Computer Music.* New Orleans, USA, 2006.

[10] Schacher. J., Neukom, M., and Bisig, D. Composing with Swarm Algorithms - Creating Interactive Audio-Visual Pieces Using Flocking Behavior. In *Proceedings of International Computer Music Conference.* Huddersfield, England, 2011.

[11] Wakefield, G., and Haru, J. *Artificial Nature: Immersive World Making.* In *Proceedings of the EvoWorkshops*. Springer, London, England, 2009.

[12] Waters, S. Performance Ecosystems: Ecological approaches to musical interaction. In *Proceedings of Electroacoustic Music Studies Network Conference.* De Montfort University, Leicester, England, 2007.

[13] http://swarms.cc (URL valid in April 2011)

[14] http://www.openframeworks.cc (URL valid in April 2011)

Implementing a Finite Difference-Based Real-time Sound Synthesizer using GPUs

Marc Sosnick William Hsu
San Francisco State University, Department of Computer Science
1600 Holloway Ave. TH 906,
San Francisco, CA, 94132, USA
msosnick@sfsu.edu whsu@sfsu.edu

ABSTRACT
In this paper, we describe an implementation of a real-time sound synthesizer using Finite Difference-based simulation of a two-dimensional membrane. Finite Difference (FD) methods can be the basis for physics-based music instrument models that generate realistic audio output. However, such methods are compute-intensive; large simulations cannot run in real time on current CPUs. Many current systems now include powerful Graphics Processing Units (GPUs), which are a good fit for FD methods. We demonstrate that it is possible to use this method to create a usable real-time audio synthesizer.

Keywords
Finite Difference, GPU, CUDA, Synthesis

1. INTRODUCTION
Most affordable desktop and laptop systems now include powerful Graphics Processing Units (GPUs). Recent GPUs from companies such as Nvidia (http://www.nvidia.com) have adopted more flexible architectures to support general purpose computing. Software support for non-graphics computing on GPUs has also improved significantly in the last few years, with environments such as Nvidia's Compute Unified Device Architecture (CUDA) [8] and OpenCL [9]. As a result, there has been much development of general computing on GPUs. In particular, we are interested in the use of GPUs for real-time sound synthesis.

In previous work, we have shown [12] that it was possible to use the computationally expensive finite difference method to generate sound in real-time. We have subsequently been working to create a usable synthesizer package, *Finite Difference Synthesizer (FDS)*, based on the finite difference method, to generate real-time sound.

Our implementation uses a finite difference-based simulation for a two-dimensional membrane [1, 7] which runs in real time on the GPU; the architecture of the GPU is particularly well suited for this type of algorithm. Finite difference methods are well known as an effective approach for sound synthesis; see for example [3, 7]. Such methods can be a framework for constructing a number of complex software percussion instruments; sound examples generated using the synthesis package will be available at http://userwww.sfsu.edu/~whsu/FDGPU. Finite difference-based sound synthesis for large or fine-grained membranes and plates is too expensive to run in real time on CPUs. Previous studies on audio processing using earlier generation GPUs and software have been mixed (see for example [14, 5]). Our earlier results [12] show that it is feasible to implement such compute-intensive real-time sound synthesis algorithms on GPUs. We have since re-designed our software framework to improve the system's use in a real-time performance setting. This paper will focus on software details of our real-time finite difference-based synthesizer for percussion instruments.

Our paper is organized as follows. Section 2 is an overview of related work on high-performance audio computing. In Section 3 we describe the finite difference synthesis algorithm we worked with. In section 4 we discuss details of our software implementation. We present experimental setup in section 5, results and measurements in Section 6. Conclusions are drawn in Section 7. Section 8 outlines possible future directions for the FDS.

2. RELATED WORK
The website http://gpgpu.org is a major clearinghouse for information on general purpose computing on GPUs. Relatively few audio-related projects are documented on the site. [14] implemented seven audio DSP algorithms on a GPU. [11] studied waveguide-based room acoustics simulations using GPUs.

GPUs have been used in the real-time rendering of complex auditory scenes with multiple sources. In [4], the GPU is used primarily for computing particle collisions to drive audio events. [16] uses the GPU for calculating modal synthesis-based audio for large numbers of sounding objects. [13] proposed a method for efficient filter implementation on GPUs, and applied it to synthesis of large numbers of sound sources in virtual environments.

Faust [10] is a framework for parallelizing audio applications and plug-ins; it does not currently support GPU computing.

Bilbao has studied extensively the use of finite differencing for sound synthesis; see for example [3]. Since large models based on finite difference methods are too expensive for real-time performance on CPUs, work has been done for example on FPGA-based implementations [7]. Our approach leverages GPUs that are already common on commodity systems, and does not require custom hardware. Preliminary results and measurements were reported in [12]; this paper focuses on details of the current software implementation.

3. FINITE DIFFERENCE AGORITHM
We use the finite difference (FD) method of approximation of the wave equation with dissipation to simulate a membrane in two dimensions as derived by Adib [1]. A square membrane is modeled with a horizontal x-y grid of points. The continuous function $u(x, y, t)$ is defined on the spatial x and y, and time t; u is the vertical displacement at the point (x, y) at time t.

The derivation of the approximation we used can be found in [3, 6, 12] and is given as:

$$u_{i,j}^{n+1} = \left[1+\frac{\eta\Delta t}{2}\right]^{-1} \left\{ \begin{array}{l} \rho\left[u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}\right] \\ +2u_{i,j}^{n} - \left[1+\frac{\eta\Delta t}{2}\right]u_{i,j}^{n-1} \end{array} \right\} \quad (1)$$

where, from [6]:

$$\rho = \left(v \cdot \frac{\Delta t}{\Delta x}\right)^2 \quad (2)$$

such that v is velocity of the wave in the medium and η is the viscosity coefficient. For our experiments, we treat η and ρ as constants, and used known stable values from Land [6], but allow these to be changed using Open Sound Control (OSC) methods in the synthesis package.

We implemented u as three 2-D matrices of single-precision (4-byte) floating point numbers so as to maintain compatibility with Nvidia devices of compute capability 1.2 or lower [8]. We use the leap-frog algorithm to calculate the values at $u_{i,j}^{n+1}$ given the values of $u_{i,j}^{n-1}$ and $u_{i,j}^{n}$ [1]. Boundary conditions are maintained at each iteration by testing the values of i and j and adjusting $u_{i,j}^{n}$ appropriately. A scalar gain value is used to either clamp the edge (boundary gain = 0) or allow motion dependent on the adjacent internal grid point times the boundary gain (boundary gain < 1) [5]. Corners are given no special consideration. To obtain different sounds, the values of n (grid size), η, ρ, and boundary gain are manipulated. For example, values of η=2x10^{-4}, ρ=0.5, n = 6, and a boundary gain of 0.75 produces a bell-like tone; values of η=2x10^{-4}, ρ=0.5, n = 16, and a boundary gain of 0 produces a drum like tone. Further examples of this can be found at http://userwww.sfsu.edu/~whsu/FDGPU.

To obtain audio output, the membrane must be excited in some fashion, roughly analogous to striking or plucking the membrane. We use a simple Gaussian impulse to initialize/excite the membrane. $u_{i,j}^{n-1}$ is set to 0, and $u_{i,j}^{n}$ to a Gaussian impulse, as suggested in [3, 6]. To obtain audio output, a point on the membrane is chosen, and the value for $u_{i,j}^{n}$ is sampled and scaled at each iteration. For the FDS, the center point of the grid is chosen as the output point.

We used Nvidia's Compute Unified Device Architecture (CUDA) extension to C to implement our parallel implementation of the finite difference simulation for the GPU. Nvidia's GPU hardware is a SIMT (single instruction multiple threads) architecture using scalable arrays of multithreaded streaming multiprocessors [8]. CUDA divides system hardware into *host* and *device*, where the host is the system (PC desktop or laptop) in which the Nvidia device (or GPU) resides, and the device is the Nvidia GPU on which the parallel program, or *kernel*, executes. The host system first prepares the device and then hands off execution of the kernels to the device. Each kernel is executed on the device in a *thread*, and threads are combined into one, two, or three dimensional *thread blocks*. In a kernel, a thread can obtain its unique *x, y, z* position in the thread block, which is what we use to determine the thread's position when calculating u. All threads in a thread block execute simultaneously, but can be synchronized [8].

Memory between the host and device can be independent or integrated with system memory, but in either case are addressed separately on the host and device. On some systems page-locked host memory (called *pinned memory*) can be mapped to the device [8]. Pinned memory simplifies and reduces the overhead of asynchronously transferring results from the device to the host.

In our parallel implementation of the FD simulation, a single thread is mapped to and calculates each FD grid point. A thread determines its position in the grid by finding its 2-D location in the thread block [8]. At each time-step, each thread calculates one update of the $u_{i,j}^{n+1}$ array. Each thread checks to see if its grid-point is at a boundary; if so, it applies the boundary condition to that point. The thread that corresponds to the output grid-point also updates the output buffer with its vertical displacement over multiple time steps. In order to maintain coherence over time, the threads are synchronized at critical points.

To execute each kernel, the host hands off execution to the GPU device. The simulation runs for several time-steps, and the output buffer is filled with the computation results, after which execution on the GPU device stops.

4. IMPLEMENTATION

Our software implementation of the finite difference membrane simulation is written in C++ using Nvidia CUDA (The package will be available for download at http://userwww.sfsu.edu/~whsu/FDGPU). The FDS system uses PortAudio (http://www.portaudio.com) (PA) for real-time audio I/O, liblo (http://liblo.sourceforge.net) for the Open Sound Control (OSC) interface.

In order to minimize data transfer latency, both the simulation data as well as the buffered audio data are stored in GPU memory. Four grids are kept in GPU memory: FD simulation grids for the current and two past time steps, as well as a Gaussian impulse that is used to excite the membrane. When an excitation command is received, a separate kernel positions, scales and copies the Gaussian impulse grid into the FD simulation grids.

Overall, an FDS-based system acts as an OSC server, waiting for OSC packets to be received, and reacting appropriately to controller input.

4.1 Multithreading

During execution, there are three simultaneous threads running on the host system (Figure 1): a primary foreground thread handling control, a Port Audio callback thread [2] for system audio output, and a thread performing the finite difference simulation producing audio data. Communication between the audio data producer (FD Engine) and consumer (PA Callback) is achieved using the PA thread-safe ring buffer.

4.1.1 Primary foreground thread

In addition to initializing and shutting down the system, the primary foreground thread handles OSC signals and sends user interface commands to the other threads through appropriate semaphores.

4.1.2 Finite Difference Thread

The finite difference simulation is contained in its own thread, and communication with the GPU occurs exclusively in this thread. As mentioned above, control of the simulator such as excitation of the membrane is triggered from the primary thread. After initialization, the finite difference simulation runs continuously, filling the ring buffer with data as space permits. To generate sound, the FD membrane must be excited (perturbed) in some fashion. An arbitrary point on the simulation membrane is used to generate audio output; for the current version of FDS, this is the center of the grid. The value of the vertical displacement of this point at each time step is placed in the audio buffer. The FD kernel (Figure 2) updates the vertical displacement of the grid for a fixed number of timesteps. The displacement of the center point at each timestep is stored into a temporary buffer in GPU memory. The temporary GPU buffer is then copied to the ring buffer in system memory.

Initially all points on the membrane are stationary and have zero vertical displacement. Upon receipt of an excitation command via OSC (e.g. a hit), the primary foreground thread

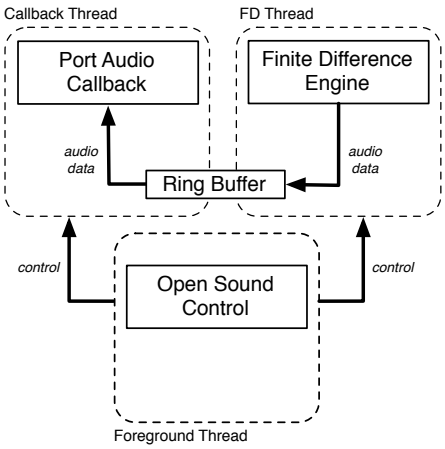


Figure 1. Thread configuration during execution.

sends a command to the FD thread to excite the membrane. In the FD thread, upon receipt of this command an excitation kernel is called (Figure 2). The excitation kernel copies the Gaussian curve stored in GPU memory to the FD membrane history buffer; this impulse induces vibration in the FD membrane. The excitation kernel can reposition the center of the Gaussian curve to approximate striking (plucking) the membrane at different locations on the surface. The Gaussian curve can also be scaled to simulate harder or softer strikes.

4.1.3 PA Callback Thread

The PA callback thread is a standard audio callback. The callback reads available data in the ring buffer and copies the necessary samples to the Portaudio audio output buffer.

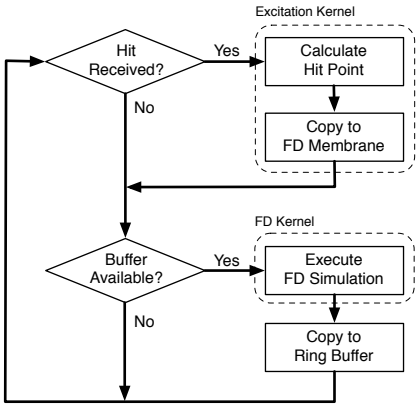


Figure 2. Main FD Thread Loop

4.2 OSC Methods

OSC methods [15] for exciting the membrane using fixed and variable positions, as well as varying amplitude, are available. In addition, FD simulation parameters can be changed using OSC methods, to simulate membranes with different material properties

As discussed in Section 3, for the FD simulation to generate different sounds, the values of n (grid size), η, ρ, and boundary gain are manipulated. For real-time performance, only some of these can be changed in real-time.

For the current implementation of the FDS, after initialization, grid size (n) remains constant. Allocation of both system and GPU memory takes too long to enable reconfiguration in real-time. Once the grid size has been set for a particular sound, it cannot be changed in real-time. The FD simulation parameters η, ρ, and boundary gain (see above) can be changed in real-time; OSC methods are provided for each of these parameters.

An OSC controller for the iPhone was developed for use in testing (Figure 3) using TouchOSC (http://hexler.net/). Touching the X-Y pad results in an excitation to the corresponding location on the FD membrane, while the *Amp* slider linearly scales this Gaussian excitation impulse. *Pulse*

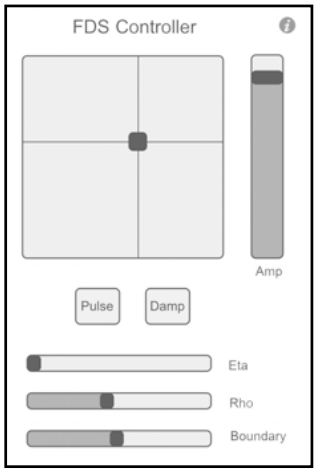


Figure 3. OSC controller interface used in testing.

and *Damp* are momentary pushbuttons; *Pulse* sends a full-amplitude Gaussian impulse to the center of the FD membrane, and *Damp* stops all FD membrane vibration. *Eta*, *Rho* and *Boundary* sliders modulate the parameters described in Section 3.

5. EXPERIMENTAL SETUP

5.1 System Configurations

We tested our system on a MacBook Pro with a 2.66 GHz Intel Core i7, 4 GB 1067 MHz DDR3 RAM, and a GeForce GT 330M GPU running OS 10.6.6.

Timings were taken for two setups. For setup I we held constant a grid size of 21x21 points, and used kernel output buffer sizes of 8, 512, and 4096 entries. For setup II we held the kernel output buffer constant at 4096 entries, and used FD grid sizes of 15x15, 18x18, and 21x21. These values were chosen to correspond to previous tests performed in [12]. In all cases, the ring buffer was guaranteed to have enough space to accept the full contents of the kernel output buffer.

5.2 Testing

For each timing measurement (i.e. each buffer size in setup I and each grid size in setup II), we repeated the following sequence 500 times: run the excitation kernel, check ring buffer space, perform the FD simulation, and copy the FD simulation output to the ring buffer. Timing measurements were averaged over these 500 runs. The built-in CUDA timer routines were used to time memory transfer, excitation, and FD membrane kernel execution times.

A separate test was run with each of the above buffer and grid configurations to ensure that the audio quality was adequate. For this test, the membrane was excited and allowed to play for one second. This was repeated five times. Any audio output buffer underruns were counted; buffer underruns would indicate poor audio quality.

Qualitative testing of the FDS was performed using the OSC controller in Figure 3, changing parameters in real-time.

6. EXPERIMENTAL RESULTS

The results for the timing tests are summarized in Table 1 and Table 2. Total time is the sum of excitation time, finite difference time, and memory transfer time. Buffer sizes of 8,

512, and 4096 samples correspond to audio output of durations 0.181 ms, 11.6 ms and 92.8 ms at a sampling rate of 44,100 Hz.

For the audio quality test, all kernel output buffer and grid configurations produced no audio output buffer underruns.

Satisfactory percussive sounds were produced using the OSC controller interface in qualitative testing. It was found that the FDS's output was sensitive to changes in the FD parameters, especially η and ρ. Recording of some of these tests will be available at http://userwww.sfsu.edu/~whsu/FDGPU.

7. CONCLUSIONS

We have successfully implemented a usable real-time audio synthesizer based on computationally expensive FD simulations. The results of the audio quality tests show that with carefully chosen parameters the FD membrane scheme can generate audio data sufficiently fast to support real-time synthesis. As expected, the majority of the processing time is spent performing the finite difference simulation.

Table 1. Setup I: Results for fixed 21 x 21 grid and varying output buffer size. Timings are averaged over 500 runs.

Buffer Size (samples)	Excitation Time (ms)	Finite Difference Time (ms)	Memory Transfer Time (ms)	Total Time (ms)
8	0.04	0.56	0.02	0.62
512	0.03	6.78	0.01	6.82
4096	0.03	34.31	0.03	34.37

Table 1 shows that as the buffer size increases, the efficiency increases. Time to calculate one sample (time per sample, where 1 sample = 0.026 ms of audio at a sampling rate of 44,100 Hz) for an 8 sample buffer is 0.078 ms, but for a 512 sample buffer it is 0.013 ms, and for a 4096 sample buffer it is 0.008 ms. This decreasing execution time makes sense as the overhead of starting and stopping the simulation and transferring the data is leveraged over a larger buffer size. But this also shows that buffer parameters must be carefully tuned in order to assure adequate real-time performance.

Table 2. Setup II: Results for a fixed buffer size of 4096 samples, and varying grid size. Timings are averaged over 500 runs.

Grid Size (points)	Excitation Time (ms)	Finite Difference Time (ms)	Memory Transfer Time (ms)	Total Time (ms)
15x15	0.03	30.26	0.03	30.32
18x18	0.03	31.81	0.03	31.87
21x21	0.03	34.73	0.03	34.37

Table 2 shows that with an increasing grid size, the simulation efficiency increases. The time to calculate each grid point is 0.13 ms for a 15x15 grid, 0.10 ms for an 18x18 grid, and 0.08 ms for a 21x21 grid.

8. FUTURE WORK

As the majority of execution time is spent in the FD simulation, improvements to this kernel would result in improvements to the overall system.

Other computationally expensive simulations may provide interesting audio results. These simulations would be particularly suited to this synthesis package if the simulation can be efficiently calculated in parallel using GPUs.

To leverage multiple processor environments, current plans include porting the GPU code to the industry-standard OpenCL language [9] and testing it across heterogeneous compute platforms

9. REFERENCES

[1] Adib, A. Study Notes on Numerical Solutions of the Wave Equation with the Finite Difference Method. *arXiv:physics/0009068v2 [physics.comp-ph]. 4 October 2000*. Downloaded from http://arxiv.org/abs/physics/0009068v2 on April 15, 2010.

[2] Bencina, R., and Burk, P. PortAudio – an Open Source Cross Platform Audio API. *Proceedings of the ICMC, 2001.*

[3] Bilbao, S. A finite difference scheme for plate synthesis. *Proceedings of the International Computer Music Conference*, pp. 119-122, 2005.

[4] van den Doel, K., Knott, D., and Pai, D. Interactive Simulation of Complex Audio-Visual Scenes. *Presence: Teleoperators and Virtual Environments*, Vol. 13, No. 1, pp. 99-111, 2004.

[5] Gallo, E., and Tsingos, N. Efficient 3D Audio Processing on the GPU. In *Proceedings of the ACM Workshop on General Purpose Computing on Graphics Processors*, August 2004.

[6] Land, B. Finite difference drum/chime. From http://instruct1.cit.cornell.edu/courses/ece576/LABS/f2009/lab4.html, 4/15/2010.

[7] Motuk, E., Woods, R., Bilbao, S., and McAllister, J. Design Methodology for Real-Time FPGA-Based Sound Synthesis. *IEEE Transactions on Signal Processing*, Vol. 55, No. 12, pp. 5833 – 5845, 2007.

[8] *Nvidia CUDA Programming Guide, version 2.3.1. 8/26/2009.* Downloaded 4/21/2010 from http://developer.download.nvidia.com/compute/cuda/2_3/toolkit/docs/Nvidia_CUDA_Programming_Guide_2.3.pdf.

[9] *Nvidia OpenCL Programming Guide, version 2.3. 8/27/2009*. Downloaded 4/21/2010 from http://www.nvidia.com/content/cudazone/download/OpenCL/Nvidia_OpenCL_ProgrammingGuide.pdf

[10] Orlarey, Y., Fober, D., and Letz, S. Parallelization of Audio Applications with Faust. In *Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference*, pp. 23-25, 2009.

[11] N. Rober, N., Kaminski, U., and Masuch, M. Ray Acoustics using Computer Graphics Technology. In *Proceedings of DAFx*, 2007.

[12] Sosnick, M., and Hsu, W. Efficient Finite Difference-Based Sound Synthesis Using GPUs. In *Proceedings of SMC Conference 2010, Barcelona*.

[13] Trebien, F., and Oliveira, M. Realistic real-time sound re-synthesis and processing for interactive virtual worlds. *The Visual Computer,* Vol. 25, No. 5-7, 2009.

[14] Whalen, S. Audio and the Graphics Processing Unit. Technical Report, Downloaded 4/21/2010 from http://www.node99.org/papers/gpuaudio.pdf.

[15] Wright, M. *The Open Sound Control 1.0 Specification Version 1.0*, March 26 2002. From http://opensoundcontrol.org/spec-1_0

[16] Zhang, Q., and Ye, L. Physically-Based Sound Synthesis on GPUs. In *Entertainment Computing - ICEC 2005, Lecture Notes in Computer Science,* Vol. 3711/2005.

An Artificial Intelligence Architecture for Musical Expressiveness that Learns by Imitation

Axel Tidemann
IDI, Norwegian University of Science and Technology
Sem Sælands vei 7-9
7491 Trondheim, Norway
axel.tidemann@gmail.com

ABSTRACT

Interacting with musical avatars have been increasingly popular over the years, with the introduction of games like Guitar Hero and Rock Band. These games provide MIDI-equipped controllers that look like their real-world counterparts (e.g. MIDI guitar, MIDI drumkit) that the users play to control their designated avatar in the game. The performance of the user is measured against a score that needs to be followed. However, the avatar does not move in response to how the user plays, it follows some predefined movement pattern. If the user plays badly, the game ends with the avatar ending the performance (i.e. throwing the guitar on the floor). The gaming experience would increase if the avatar would move in accordance with user input. This paper presents an architecture that couples musical input with body movement. Using imitation learning, a simulated human robot learns to play the drums like human drummers do, both visually and auditory. Learning data is recorded using MIDI and motion tracking. The system uses an artificial intelligence approach to implement imitation learning, employing artificial neural networks.

Keywords

Modeling Human Behaviour, Drumming, Artificial Intelligence

1. INTRODUCTION

The ubiquity of cheap processing power and new physical interfaces has led to the introduction of novel applications when it comes to expressive music performance in the digital realm. Although computers have been used for musical purposes for decades, they have become more prominent in popular culture with the introduction of games like Guitar Hero[1] and Rock Band[2]. In these games, the user plays along with a score displayed on the screen. The user performs with MIDI interfaces that look like real instruments, such as a guitar[3] or a drum kit. As part of the game, animated musicians play the different musical instruments in the song. However, these animated musicians (or avatars) do not move in accordance with the user input. If the user makes an error, it is not reflected in the behaviour of the corresponding avatar. The only way the avatar reacts to the input of the user is if the user performs poorly to the extent that the game is terminated before the song is over; the avatar subsequently throws the guitar on the floor.

These games would greatly benefit from some way to move the corresponding avatar in accordance with user input, with natural movement as a result. This would enhance the gaming experience. This paper presents an architecture that uses learning by imitation to move a simulated robot based on musical input. The system learns to play drums like human drummers do. The architecture is divided into two subsystems; a sound system that imitates the playing style (i.e. it sounds like a human drummer) and a motor system that generates the corresponding arm movements. Both systems use imitation as the learning principle. By seeing and hearing human drummers, the system is able to imitate their playing style. Why use imitation as the learning mechanism? First of all, this is a way that humans transfer motor knowledge between individuals. The ability to imitate is without a doubt a cornerstone of human society. Secondly, when trying to make a machine learn a human quality such as musical expressiveness, it makes sense to use the same mechanism as that of humans. Instead of trying to formulate human behaviour using mathematical formulas, it is more intuitive to simply *show* the machine what it should do. Furthermore, learning by imitation implies an internalization (i.e. a *model*) of the acquired knowledge. An artificial drummer that merely plays back a recording is not of great interest, neither expressively nor research-wise. The machine uses a learned model to generate new music, that will be *similar* to the original, but not *identical*. These are the main reasons imitation learning is employed in the architecture, which uses an artificial intelligence approach to implement imitation learning.

2. BACKGROUND: IMITATION LEARNING

Imitation learning has been extensively studied in psychology and is considered an important part of human society [17, 14]. The discovery of *mirror neurons* was considered as a possible "neural candidate" for the imitative capability in the human brain [19]. Mirror neurons were found to be active both during observation and production of the same movement. The mirror neurons were also hypothesized to be the neural mechanism behind empathy, allowing humans to transform their viewpoint into that of others [5]. However, recent studies have questioned the comparison between a mirror neuron system in monkeys and humans [12]; mirror neurons remain controversial.

In the artificial intelligence community, imitation learning has gained momentum as a way to program desired behaviours in robots. Schaal [21] suggests model-based ap-

[1] hub.guitarhero.com
[2] www.rockband.com
[3] Fender released a real guitar on March 1st, 2011 that can be played as a controller for Rock Band.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

proaches as the best way to implement imitative behaviour; this consists of pairing an inverse model (controller) with a forward model (predictor), an approach that stems from control literature [11]. Wolpert et al. argue that such inverse/forward couplings are present in the cerebellum [28], leading to an architecture based on those principles. Demiris et al. have also investigated an imitative architecture based on such inverse/forward pairings [4]; there are some fMRI studies suggesting such an ordering is present in the brain [9].

There are other modular architectures for imitation learning that take a slightly different route by defining modules for different stages of sensorimotor processing, such as perception, recognition and action selection [6, 13]. Some researchers focus solely on neural network architectures designed for imitation learning [22, 2, 1].

In music, it is evident how humans imitate others when learning to play instruments. In the cross section between music technology, machine learning and music performance are systems that focus on capturing human expressiveness. Saunders et al. [20] use string kernels as a classification method for pianists. The string kernels are used to modify changes in tempo and velocity when playing a classical piece of music. Tobudic and Widmer use first-order logic to describe the same changes [26], the system can subsequently be used to classify pianists based on their playing style. Case-Based Reasoning (an artificial intelligence method where known solutions to old problems are re-used to find solutions to new problems) have been used to model human expressiveness, such as mood [3] and how the tempo can change, but still maintain the original sentiment [7]. Pachet [16] has a system called "The Continuator" that employs Hidden Markov Models to predict the next note; this is a real-time system that can be used to interact with other musicians. Raphael [18] has a system that allows a soloist to practice along with a computer playing a score; the system learns how the soloist varies the tempo over time, and plays along with the tempo drift. In the music software industry, sophisticated drum sample software (e.g. FXpansion BFD, Toontrack EZdrummer, DigiDesign Strike, Reason Drum Kits, Native Instruments Battery) contain gigabytes of samples, but no *intelligent* way of creating human-like drum tracks, apart from adding random noise that is to be perceived as human. The research in this paper addresses this issue.

3. ARCHITECTURE

The architecture that implements the artificial drummer is called "Software for Hierarchical Extraction and Imitation of Drum Patterns in a Learning Agent" (SHEILA). It is comprised of two subsystems, a sound system that imitates the playing style (i.e. what you can *hear*) and a motor system that imitates the corresponding motor actions (i.e. what you can *see*). How the two subsystems interact can be seen in figure 1. This separation reveals a simplification: the sound system can be used as a groovy drum machine by itself, since it outputs the imitated sound. The motor system generates the corresponding arm movements of the drummer. This separation was made for two reasons: development-wise, it was easier to make a clear division between sound and motor actions. Secondly, this frees up the necessity of simulating physical drums as well. The artificial drummer will move its arms in accordance with the sound that is produced, however the movement of the arms does not generate sound by hitting a drum. If the sound were to be generated by the arm movement, the problem would be vastly more complex, requiring a model of physical drums.

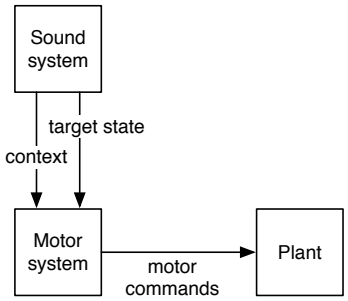


Figure 1: A simplified overview of the architecture: the sound system produces sound signals, as well as driving the motor system. The motor system issues motor commands to achieve the movements implied by the sound signals.

For an on-screen avatar this is not necessary - for the end user of the system, the movement and sound from the artificial drummer will be realistic. The different subsystems will now be presented.

3.1 The Sound System

The sound system learns user-specific variations from human drummers. An important aspect of human drumming is the introduction of *variations*. The drummer can play small-scale variations, e.g. varying the velocity (how hard a note is played) and timing (how much the note is before or after the metronome). The drummer can also add large-scale variations, such as altering the pattern played altogether. This is often referred to as a *break*, something the drummer does for rhythmic and dramatic effect, adding dynamics to a song. The small- and large-scale variations add up to the *groove* of the drummer, which is what the sound system imitates.

MIDI recordings of human drummers provide data that the system is trained on. Drum patterns are analyzed in a hierarchical manner: the MIDI drum sequence is transformed into a string. Similar patterns are found in the string by searching for supermaximal repeats, a method used to search for sequences in genes [8]. This method allows similar patterns to be extracted from the MIDI stream. The patterns are used to train Echo State Networks (ESNs) [10], a neural network architecture characterized by its huge memory capacity and fast training algorithm. These ESNs are not driven by input, they are self-generating networks; the networks use feedback connections from the output layer to reverberate in the desired state. The ESNs can be thought of as having a pulse that generates the desired groove after learning. More details can be found in [23].

3.2 The Motor System

The motor system is responsible for the imitation of arm movements. The approach is to pair an inverse model (a controller) with a forward model (a predictor), an approach well known in robot control literature [11]. The motor system uses several such pairs of inverse and forward models. The motor system is in turn inspired by two other architectures for motor control and learning that use multiple paired inverse and forward models [28, 4]. See [25] for more details.

3.3 Combining the Motor and Sound System

To create an animated artificial drummer that both sounds and looks like a real drummer, the two subsystems are connected to provide sound and animation. The output of the

sound system is used to create the sound, but also to *drive* the motor system. The actual sound output is used as the *desired state* for the motor system. The inverse model receives signals that describe what the *end result* of the movement should be. This sound signal is in a different coordinate system than that of the current state of the motor system, which makes it harder for the inverse model to learn the corresponding relationships.

4. EXPERIMENTAL SETUP

In order to train the system, five human drummers were told to play specific patterns along with a song written by the author. The drummers could then introduce large-scale variations as they saw fit. MIDI was recorded using a velocity sensitive electronic drumkit, the Roland TD-3. Motion tracking was done with a Pro Reflex tracking system. Pro Reflex makes use of infrared cameras to track position of fluorescent markers over time. Using motion tracking effectively solves the correspondence problem [15], since the recorded 3D coordinates could be mapped directly to the artificial drummer. The robot arm was implemented as a four degrees of freedom (DOF) model based on the human arm [27] (a 3DOF spherical shoulder joint, 1DOF revolute elbow joint). The entire robot was described by 8DOF.

5. DISCUSSION

After the recording and training of the system, SHEILA was used to imitate the playing style of the human drummers that served as teacher. By performing statistical analysis on the resulting drum patterns, it was revealed that the imitated drum patterns are similar, but not identical. Further detailed results of the sound system can be found in [23].

The performance of the motor system was also very good. When comparing recorded training data with performance data, the error was less than 0.05%. The motor system relies heavily on biological properties such as self-organization during learning; it is an AI architecture for motor control and learning in itself. The self-organizing properties have been thoroughly investigated elsewhere, see [25, 24]. An example of the imitative capabilities can be seen online[4].

However, the focus of this paper is how this combination of AI subsystems can be used for musical expressiveness, and in particular in games like Guitar Hero and Rock Band. Why use a computationally expensive artificial intelligence approach, instead of simply playing back a recording of the desired behaviour? First of all, such an approach would yield an identical result each time it is used. By employing imitation learning, the generated drum patterns will sound similar, but not identical. Furthermore, in order to truly imitate human movement, it is imperative that the underlying approach is biologically inspired. For this reason, the research in this paper is multi-disciplinary; it focuses on imitation of musical expressiveness using artificial intelligence mechanisms that can faithfully reproduce this human behaviour.

The sound system was designed around a more pragmatic, hierarchical approach. However, it was implemented using Echo State Networks, which are modeled on the neural networks present in our brains. In order to implement a human quality such as groove, it makes sense to implement this capability using a biologically inspired method.

The motor system was more directly inspired by existing neuroscientific models of how the brain operate [28]. Motor control and learning have been a focal point for AI research for decades; an architecture that is to implement this ability would benefit from an approach based on neuroscientific principles. The research in this paper was done on a simulated robot, since a real robot with the agility equal to humans is prohibitively expensive. However, one can envision that in the future robot technology will be cheaper and with greater dexterity. The architecture could then be employed on a real robot, since its design is based on robot control mechanisms [11]: the continuous outputs of the inverse models (i.e. Echo State Networks) could easily be converted to voltages used to drive a real robot.

A key element is that the architecture is in principle independent of what kind of instrument it is supposed to imitate. Both the sound and motor system are independent of the drumming domain. As long as there is some repetitive melodic structure (e.g. guitar riffs and bass lines), the sound system can model it. Motion tracking can be used on various parts of the body. Why was drumming chosen as the application? There are two main reasons: 1) Playing drums is very repetitive, where the pattern is normally reproduced every bar. For melodic instruments, the repeated pattern (i.e. melody) can last longer. The makes it easier to learn models of a particular playing style, and made for a good starting point when exploring this research path. 2) Imitating the movement of the drummer could be limited to the arms only. Granted, the drummer invariably moves the entire body, however the arms will provide a sufficient subset of the body movement in order to imitate a playing style, since a drummer is stationary during playing. The movement of the arms is also easy to visualize. In the case of guitarists, the playing style to be imitated can sometimes involve more of the entire body. Extreme examples are the particular walk of AC/DC guitarist Angus Young, Jimi Hendrix playing the guitar behind his back, or The Who's guitarist Pete Townshend who plays the guitar with a "windmill" motion. These are prime examples of the possibility to imitate the playing style of guitarists.

Given the independence of SHEILA regarding which instrument to imitate, it could be employed in imitative settings in other applications. When it comes games like Guitar Hero and Rock band, two possible ways of implementing the architecture could be envisioned: first, musicians on screen that are *not* controlled by humans could be implemented using SHEILA. The whole point of these games is to give the illusion of playing in a live rock band. If all the other computer controlled characters were implemented using SHEILA, their performance would be slightly different each time, but still recognizable. No human musician plays a musical piece exactly the same way twice, so this would greatly add to the feeling of realism of playing along with other characters. Secondly, it could be envisioned that human players wanting to control the on screen musicians could take the place of the sound system. The input of the player would then drive the motor system, so the on screen musician would move in response to the player's input, but would still look like the original musician. For instance, if the player is controlling Lars Ulrich of Metallica, the sound of Ulrich playing would correspond to the performance of the player, but still *look* like how Ulrich would play it. The input from the user would most likely not be identical to that of Ulrich himself, but an advantage of employing neural networks is their ability to generalize and handle noisy situations, which would deal with these kinds of situations. An important aspect of employing the SHEILA architecture would be the cost: using motion capture is an expensive process. However, motion capture is already being used for the creation of such games[5], so the cost issue in this regard

[4] www.idi.ntnu.no/∼tidemann/sheila/SHEILAweb.mov

[5] www.usatoday.com/tech/gaming/2008-12-14-metallica-game-qanda_N.htm, retrieved 2011-02-04

would not be prohibitive. To conclude, SHEILA has so far shown promising results regarding its ability to imitate human musical expressiveness, and would be a good approach to enhance games like Rock Band or Guitar Hero.

6. FUTURE WORK

Parts of this paper have been focusing on how this research can be applied in commercially available applications. An open source program called *Frets on Fire*[6] could serve as the starting point for developing SHEILA in a game similar to Rock Band or Guitar Hero.

Although the architecture has shown good results when it comes to imitation of known patterns, the next step will be to examine whether it can generalize and play *new* patterns that have not been part of the training data. This can be tested by recording different patterns from a human drummer, and training the system on selected patterns. The artificial drummer could then be told to play a novel pattern that the system has not been trained on. The output of the system could then be matched against how the teacher drummer would actually play this pattern.

7. REFERENCES

[1] A. Billard and G. Hayes. DRAMA, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, 7(1):35–63, 1999.

[2] A. Cangelosi and T. Riga. An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots. *Cognitive Science*, 30(4):673–689, 2006.

[3] R. L. de Mantaras and J. L. Arcos. AI and music from composition to expressive performance. *AI Mag.*, 23(3):43–57, 2002.

[4] Y. Demiris and B. Khadhouri. Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and Autonomous Systems*, 54:361–369, 2006.

[5] V. Gallese and A. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12), 1998.

[6] P. Gaussier, S. Moga, J. P. Banquet, and M. Quoy. From perception-action loops to imitation processes: A bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, 1(7):701–727, 1998.

[7] M. Grachten, J. Arcos, and R. de Mantaras. A case based approach to expressivity-aware tempo transformation. *Machine Learning*, 65(2):411–437, 2006.

[8] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology.* Cambridge University Press, New York, NY, USA, 1997.

[9] H. Imamizu, T. Kuroda, T. Yoshioka, and M. Kawato. Functional magnetic resonance imaging examination of two modular architectures for switching multiple internal models. *Journal of Neuroscience*, 24(5):1173–1181, 2004.

[10] H. Jaeger and H. Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80, 2004.

[11] M. I. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354, 1992.

[12] A. Lingnau, B. Gesierich, and A. Caramazza. Asymmetric fMRI adaptation reveals no evidence for mirror neurons in humans. *Proceedings of the National Academy of Sciences*, 106(24):9925–9930, 2009.

[13] M. J. Matarić. *Imitation in animals and artifacts*, chapter Sensory-Motor Primitives as a Basis for Learning by Imitation: Linking Perception to Action and Biology to Robotics, pages 392–422. MIT Press, Cambridge, 2002.

[14] A. N. Meltzoff and M. K. Moore. Imitation of facial and manual gestures by human neonates. *Science*, 198:75–78, October 1977.

[15] C. L. Nehaniv and K. Dautenhahn. *Imitation in Animals and Artifacts*, chapter The Correspondence Problem, pages 41–63. MIT Press, Cambridge, 2002.

[16] F. Pachet. Interacting with a musical learning system: The continuator. In *ICMAI '02: Proceedings of the Second International Conference on Music and Artificial Intelligence*, pages 119–132, London, UK, 2002. Springer-Verlag.

[17] J. Piaget. *Play, dreams and imitation in childhood.* W. W. Norton, New York, 1962.

[18] C. Raphael. Orchestra in a box: A system for real-time musical accompaniment. In *IJCAI workshop program APP-5*, pages 5–10, 2003.

[19] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.

[20] C. Saunders, D. Hardoon, J. Shawe-Taylor, and G. Widmer. Using string kernels to identify famous performers from their playing style. *Intelligent Data Analysis*, 12(4):425–440, 2008.

[21] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

[22] J. Tani, M. Ito, and Y. Sugita. Self-organization of distributedly represented multiple behavior schemata in a mirror system: Reviews of robot experiments using RNNPB. *Neural Networks*, 17:1273–1289, 2004.

[23] A. Tidemann and Y. Demiris. Groovy neural networks. In *18th European Conference on Artificial Intelligence*, volume 178, pages 271–275. IOS press, July 2008.

[24] A. Tidemann and P. Öztürk. Using multiple models to imitate drumming. In *Robotics and Applications*, IASTED Technology Conferences, pages 443–452. ACTA Press, 2010.

[25] A. Tidemann, P. Öztürk, and Y. Demiris. A groovy virtual drumming agent. In *Intelligent Virtual Agents*, volume 5773 of *Lecture Notes in Computer Science*, pages 104–117. Springer Berlin / Heidelberg, 2009.

[26] A. Tobudic and G. Widmer. Learning to play like the great pianists. In L. P. Kaelbling and A. Saffiotti, editors, *IJCAI*, pages 871–876. Professional Book Center, 2005.

[27] D. Tolani and N. I. Badler. Real-time inverse kinematics of the human arm. *Presence*, 5(4):393–401, 1996.

[28] D. M. Wolpert, R. C. Miall, and M. Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9), 1998.

[6] fretsonfire.sourceforge.net

TweetDreams: Making music with the audience and the world using real-time Twitter data

Luke Dahl
CCRMA - Stanford University
660 Lomita Ct.
Stanford, CA 94305
lukedahl@ccrma.stanford.edu

Jorge Herrera
CCRMA - Stanford University
660 Lomita Ct.
Stanford, CA 94305
jorgeh@ccrma.stanford.edu

Carr Wilkerson
CCRMA - Stanford University
660 Lomita Ct.
Stanford, CA 94305
carrlane@ccrma.stanford.edu

ABSTRACT

TweetDreams is an instrument and musical composition which creates real-time sonification and visualization of tweets. Tweet data containing specified search terms is retrieved from Twitter and used to build networks of associated tweets. These networks govern the creation of melodies associated with each tweet and are displayed graphically. Audience members participate in the piece by tweeting, and their tweets are given special musical and visual prominence.

Keywords

Twitter, audience participation, sonification, data visualization, text processing, interaction, multi-user instrument.

1. INTRODUCTION

Increasing amounts of public social interaction takes place through computer networks. We share jokes, stories, and news, as well as music. Yet these online interactions take place at a distance, separated by screens and transmission delays, whereas music was originally a communal activity amongst people located together in time and space.

TweetDreams is a composition and software instrument which uses real-time data from the microblogging website Twitter[1] to bring co-located performers and audience members into a public and communal musical interaction. Tweets are pulled from Twitter's web server, displayed graphically, and sonified as short melodies. The audience, when enabled with portable computing devices and Twitter accounts, become participants in the piece. They are encouraged to tweet during the performance, and within moments of doing so their words become part of the piece for all present to see and hear.

The overall structure of the piece is controlled by the performers. They interact with the software and modify parameters to control which tweets are retrieved and how they are musically and graphically rendered.

The audience and performers knowingly participate in *TweetDreams*. Yet anyone in the world tweeting during a performance may become an unwitting musical collaborator as their tweets become part of the musical conversation.

[1] `http://twitter.com`

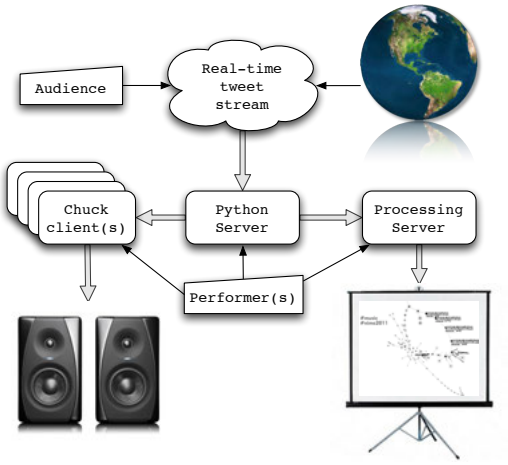


Figure 1: *TweetDreams* interaction overview

TweetDreams is implemented as three main software components: tweets are retrieved from Twitter's servers and processed by a Python application. Tweet melodies are computed and rendered using ChucK. And graphical display is rendered in Processing. Communication between all sub-systems occurs via OSC.

TweetDreams was first performed at the Milano Torino International Music (MiTo) Festival in September 2010, and has been been performed at CCRMA events a number of time since. In each case the performers have been some subset of the authors.

2. BACKGROUND AND PREVIOUS WORK

2.1 Audience Participation

The development of *TweetDreams* began with the desire to include the audience as participants in the music-making process. Audience participation in audio-visual performances has been addressed previously in a variety of ways. The audience's role may be passive yet essential, as in Levin's *DialTones*, where the music consists of the choreographed ringing of cell phones in the audience [11].

Tanaka et al. [12] discuss networked systems that present a *shared sonic environment* where participants are simultaneously performers and audience members. These systems provide simple yet powerful interfaces for creating or modifying sounds, and specific musical knowledge is not required. Barbosa presents a survey of networked digital systems for sonic creation [2].

In Freeman's *Glimmer* [5] there is a clear distinction between performers and audience. The performers are one part of a "continuous feedback loop" consisting of audience activities, video cameras and software algorithms. The au-

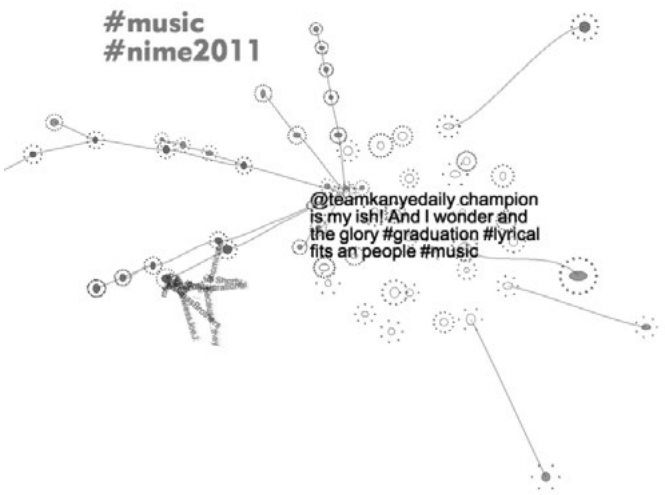


Figure 2: An example of the visualization. A new tweet, and the collapsing text of recently triggered *echos* are visible. *Search terms* are displayed in the upper left.

dience provides feedback via light-sticks that are analyzed by a digital vision system, which then provides instructions to the orchestral performers.

Typically the audience participates synchronously, that is during performance. However they may also participate before a performance takes place by submitting audio-visual materials to be used in the piece, as in *Converge* by Oh and Wang [9] or *Mad Pad* by Kruge [7].

2.2 Multi-user instruments

Jordà [6] uses the term *multi-user instruments* to describe an instrument that is performed simultaneously by multiple people, and lists examples from Cage to the *the reacTable**. He differentiates these from such concepts as *net-music* and *distributed musical systems*, and asks that we imagine instead "a hypothetical acoustic instrument that would invite many simultaneous performers".

Blain and Fels discuss *collaborative musical experiences* [3], and identify a highly restricted musical control as "the main factor common to the design of most collaborative interfaces for novices", allowing anyone, independent of their knowledge of music or the instrument, to participate. They note that with such instruments often the "overall experience takes precedence over the generation of music itself".

In many pieces for laptop or mobile phone orchestra the distinction between an ensemble of instruments and a single multi-user instrument becomes vague. The instrument may be fully distributed as in *SoundBounce* by Dahl, where network messages are used to pass sounds from one mobile phone to another [4], or centralized as in Herrera's *interV* in which a server sends performance instructions to each phone [8].

For *TweetDreams* we wanted to use mobile computing devices as a means to bring the audience into the piece. However, we found that the diversity of devices and operating systems made it prohibitively difficult to distribute a software instrument directly to all audience members. It is similarly difficult to make a web-based instrument which works on all mobile browsers. Due to these limitations we decided to use a pre-existing system to let the audience communicate with the instrument. Twitter, the popular service in which users broadcast short messages in real-time, seemed appropriate.

3. *TweetDreams* ARE MADE OF THESE

How does one make music from data that was originally created as textual statements in a natural human language? One approach would be to interpret the text of each tweet as a code or musical score, and map letters or words directly into musical notes, as in the approach taken by Alt [1]. However this would encourage the audience to compose messages that "play" this mapping, leading to tweets that are no longer idiomatic to natural language. Another approach would be to try to interpret the "meaning" of each tweet, and use that to change musical parameters (e.g. tweets could be given different sonifications based on their emotional valence). While we find this interesting, it is also quite challenging. We chose a different approach.

The music and graphics in *TweetDreams* is based on the idea of *association*. Tweets are grouped into graphs of related tweets, and associated tweets are given similar melodies and linked graphically. By this mechanism the meaning of a particular tweet does not lie in its text per se, but rather in its *network of relationships* to other tweets.

3.1 Associating tweets

The software works as follows: The system queries Twitter for any tweets containing a number of pre-defined *search terms*. One is designated as the *local search term*, and is used to recognize tweets from the audience and give them musical and graphic prominence. The others are *global search terms*, and are used to find tweets from anywhere in the world.

Each incoming tweet becomes a node in a tree-like data structure, where similar tweets are grouped together. When a new tweet arrives it is compared with all previous tweets. If it is sufficiently similar to a previous tweet it becomes the child of that tweet, and the melody for the new tweet is calculated as a mutation of its parents' melody. If a tweet is not similar to any previous tweets it becomes the root node of a new tree, to which subsequent tweets may be added.

The melody of the new tweet is then played at the same moment that its text appears along with a graphical representation of its place in the tree. After a short delay the new tweet's parent *echos*, displaying its text and playing its melody, though now acoustically and graphically attenuated. This cascade of gradually quieter and smaller echos continues up the tree, creating a rippling musical texture of related melodies.

3.2 Music

3.2.1 Calculating Melodies

Each tweet has a melody which is derived from the melody of its parent. A melody consists of six time-steps, each of which may contain a note. A note is specified as a scale degree. The new melody is constructed by a series of random mutations to the parent, where the possible mutations are *transposition* and *swapping*. In transposition a time-step is chosen randomly and the note at that time-step is transposed by a random number of scale degrees. For swapping two time-steps are chosen randomly and their note values are swapped. A total of five mutations are applied, each chosen randomly from transposition or swapping. A mutation may have no effect if the time-steps affected contain no notes, or if a swap occurs between a time-step and itself.

After mutation three checks are performed to insure the melody is well-formed. If the pitch range of a melody is too great it may not be heard as a single auditory stream, so these melodies have their range compressed. Melodies which after many mutations have become too high or too low in pitch are octave-shifted towards the center. Lastly, melodies

are shifted in time so that the first time-step contains a note. This simple algorithm leads to a nice amount of variation and similarity between parent and child melodies, creating a distinct family of melodies for a given tree and achieving the desired affect that associated tweets sound similar.

3.2.2 Music Parameters

Any new tweet which is not similar to a previous tweet becomes the root node for a new tree, and its melody is chosen from a set of pre-composed melodies. Root nodes are assigned values for a number of parameters which control how the melody is performed, and subsequent tweets which join the tree inherit these values. The performers control the sonic direction of the piece by choosing which melodies and parameters will be used for new root nodes.

Melodies are synthesized by a simple wavetable synthesizer with a low-pass filter and envelope. The related parameters are `WavetableNumber`, `FilterCutoff`, `FilterQ`, `EnvelopeAttack` and `EnvelopeDecay`. The `Mode` parameter maps scale degrees to specific pitches.

Other parameters control temporal aspects of melody performance: `StepTime` sets the duration of each time-step; `FirstEchoTime` sets the delay between triggering a new tweet's melody and its parent's melody; and `EchoTime` sets the delay between subsequent echos.

Each tweet's auditory spatialization is determined by its `Pan` parameter which is a small deviation from its parent's, creating trees which gradually spread as they grow. The number of reproduction channels can be varied according to the performance venue.

3.3 Server

The Python server (Figure 1) is in charge of handling incoming tweets, adding them to the corresponding tree, and dispatching the necessary information to the visualization and sonification sub-systems.

Incoming tweets are first classified into one of two categories (*local* or *global*) and then appended to the corresponding queue. The queues act as buffers and allow the performers to control the rate at which tweets are displayed and sonified, thereby controlling the "density" of the piece.

Cosine similarity is used to compute the distance between tweets. A Porter stemmer [10] is used to preprocess tweets to account for similar words.

3.3.1 Server Controls

Performers are able to modify the following parameters in the server: i) **Dequeueing rate**: modifies the rates at which the tweets are dequeued and dispatched; ii) **Search terms**: adds or removes search terms; iii) **Distance threshold**: changes the minimum distance required to associate tweets and thus the rate at which new trees are created.

3.4 Graphics

Tweets are displayed both as text and as a 3D graphical representation of the relationships between tweets. The visualization was created in Processing and uses OpenGL rendering to take advantage of hardware acceleration. Each tweet is represented as a circular node surrounded by a number of small "satellites" according to the number of words in the tweet. Links between connected tweets are displayed as slowly moving, slightly animated splines to convey a feeling of liveliness. Alpha transparency is used to reduce occlusion between objects.

The nodes and their links create graphically the trees of associated tweets, and virtual physics is used to animate them. Each tweet node is assigned a mass and a charge, and each node is connected to its parent node by a virtual spring. The charge causes repulsion between nodes and the mass gives them inertia. The springs counteract the repulsion, leading to trees which radiating outwards in all directions. Root nodes are connected by springs to an invisible center point. It is a dynamic system which self-organizes each time a new tweet arrives. Special attention was paid to the physical parameter values in order to avoid instability. The *Traer.Physics 3.0* library[2] was used to implement this 3D force directed layout.

Along with the node representation, the actual text of a tweet is displayed whenever a new tweet arrives or is echoed after the arrival of a new tweet. A differentiating color is used for the first appearance of a tweet. Subsequent echos of the same tweet will display the text again, but with a color scheme that differentiates local tweets from global tweets.

Throughout the performance, the search terms are displayed at the upper left of the screen, reminding the audience of the local search term they must include for their tweets to make it into the piece.

3.4.1 Graphic Controls

Performers are able to modify certain graphics parameters in real-time, to help create visual effects. The parameters are: i) **3D navigation**: moves the camera through the scene; ii) **Link length**: changes the spring constant of connecting springs, which affects the distance between connected nodes, creating the effect of visual "explosions" or "implosions"; iii) **Text size**: makes it possible to adjust the text size on the fly, to ensure that text is readable in spite of the zoom level; iv) **Trace**: controls the transparency of previous visual frames, and allows for a tracer effect; v) **Global gravity**: adds a downwards gravity which counteracts the tendency of trees to radiate.

4. PERFORMANCE

4.1 Form

TweetDreams is not automated. The performers shape the piece by controlling which search terms are used to retrieve tweets, the rate at which new tweets appear, the tendency to create new trees or build on existing trees, the melodies, timbres, and temporal character of tweet sonification, and the physics and perspective of the graphical display.

Although details differ for each performance based on audience involvement and the random nature of the world's tweets, the same basic form has been used each time:

i) **Intro**: Performers begin by tweeting an invitation to the audience to join the piece. Only the local search term is active, keeping the event density low and allowing the audience to easily see their tweets. The timbres are simple, and the graphics are zoomed to a distance that allows a sense of space; ii) **Development**: The world is brought in by adding search terms, and the event density is increased. Musical timbres become more diverse. Once the density is too high to visually track individual tweets, the camera and physics are manipulated to "explore the space"; iii) **Finale**: Search terms are removed and the dequeueing rate decreased until only new local tweets are allowed. Tweet melodies are attenuated until only the reverberated sound is heard, and the camera zooms out to reveal the full constellation of tweets that made up the piece.

4.2 Critique

A short survey was conducted of people who attended a performance of *TweetDreams* in order to understand their

[2] `http://murderandcreate.com/physics/`

experience of participating. About half the respondents reported that they were unable to interact with the piece due to not having either an internet connected device or a Twitter account. Many expressed a desire to participate, and suggested we provide additional means of input such as SMS text messaging.

Some expressed concern about the appearance of offensive content in tweets that made it into the piece. This can be addressed by implementing filtering in the server. Another concern is that while participating in the piece one is also broadcasting tweets to any Twitter *followers* who might be annoyed by the barrage of messages that make no sense outside the context of the performance.

Those who did tweet were engaged in the process of looking for and tracking their own tweets. Some reported that this required so much attention that they could not appreciate what was happening on a larger scale (a variation of Blain and Fels' claim that the overall experience occludes the music itself.) Respondents reported that the instrument responded rapidly to their messages.

Another issue was the visibility of text. Effort was made to keep tweets readable, however as the density of the piece increases it becomes difficult to find one's tweets on screen.

Some people commented on the sonification process, and felt that tweet sounds were too similar. They suggested we map words or letters to pitches to create more variety. We discussed in section 3 why we did not choose this approach, but it raises a point about the nature of this instrument. Audience members do not *play* the instrument in the sense of directly controlling what sounds are made, however their actions trigger musical and graphical events whose details are determined by their actions. It is not necessary that they entirely comprehend the mapping, and we consider this part of the piece's aesthetics.

It seems there are two ways to experience *TweetDreams*. As a participant one engages directly in the communal musical event that is transpiring and provides the materials for it. In this way the piece is *audience-mediated.* However it is also possible to passively enjoy the piece: one can sit back and voyeuristically watch the conversations of the world become music. From this perspective it functions as a type of data sonification. As an anecdote of this capability, at times during rehearsal we became aware of news events or trending topics due to the large tweet trees and similar melodies they generated.

5. FUTURE WORK AND CONCLUSION

Given the audience feedback and critiques discussed above, as well our experience performing the piece, we are considering the following improvements to make *TweetDreams* even more engaging:

i) **Implement better algorithms for calculating association.** For example, being able to derive emotions or other forms of meaning from tweets will allow the system to build more natural associations between tweets. ii) **Use current discussions to add search terms.** Currently the performers decide beforehand which search terms are used in a performance. The instrument would be more flexible if terms could be added during the piece in response to audience tweets. It would be interesting to semi-automate this process, so that new terms are automatically derived from dominant topics in recent tweets. iii) **Increase readability of tweets.** As mentioned in section 4.2, under some conditions tweet visibility is not optimal. New techniques for sizing and distributing tweet text need to be explored. iv) **Add echos up and down the graph.** Currently echos travel up trees. More complex sound textures could be achieved if echo sequences travel in all directions through the graph. v) **Use geo-location.** Twitter provides geo-location data for tweets (if the user allows) which could be incorporated in the piece. vi) **Make an installation version.** The piece was conceived as a performance, but with modifications it could be made into an installation. It could also become an interactive web-based piece, but this would require significant implementation changes.

TweetDreams is a multi-user instrument, a performance piece that invites audience participation, and a sonification and visualization of Twitter data. More significantly, it is a way to bring people who are co-located and spread across the globe into a real-time, communal and public music-making experience. The *instrument* was conceptualized with this goal in mind, but it was also designed to be experienced as a *composition*, possessing an aesthetic unity achieved through the organizing principle of association between tweets.

6. REFERENCES

[1] F. Alt, A. Sahami Shirazi, S. Legien, and A. Schmidt. Creating Meaningful Melodies from Text Messages. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, NIME 2010, pages 63–68, Sydney, Australia, 2010.

[2] A. Barbosa. Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation. *Leonardo Music Journal*, 13:53–59, 2003.

[3] T. Blaine and S. Fels. Contexts of collaborative musical experiences. In *Proceedings of the 2003 conference on New Interfaces for Musical Expression*, NIME 2003, pages 129–134, Singapore, Singapore, 2003. National University of Singapore.

[4] L. Dahl and G. Wang. Sound Bounce: Physical Metaphors in Designing Mobile Music Performance. In *Proceedings of the 2010 conference on New Interfaces for Musical Expression*, NIME 2010, 2010.

[5] J. Freeman. Large audience participation, technology, and orchestral performance. In *Proceedings of the International Computer Music Conference*, ICMC, 2005, Barcelona, Spain, 2005.

[6] S. Jordà. Multi-user Instruments: Models, Examples and Promises. In *Proceedings of the 2005 conference on New Interfaces for Musical Expression*, pages 23–26, 2005.

[7] N. Kruge and G. Wang. MadPad: A Crowdsourcing System for Audiovisual Sampling. In *New Interfaces for Musical Expression*, NIME 2011, Oslo, Norway, 2011.

[8] J. Oh, J. Herrera, N. Bryan, L. Dahl, and G. Wang. Evolving the Mobile Phone Orchestra. In *Proceedings of the 2010 conference on New Interfaces for Musical Expression*, NIME 2010, 2010.

[9] J. Oh and G. Wang. Audience-Participation Techniques Based on Social Mobile Computing. In *International Computer Music Conference*, ICMC 2011, Huddersfield, Kirklees UK, 2011.

[10] M. F. Porter. *An algorithm for suffix stripping*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[11] M. Sheridan. Composer Calling: Cell Phone Symphony Premiers. *NewMusicBox*, October 2001.

[12] A. Tanaka, N. Tokui, and A. Momeni. Facilitating Collective Musical Creativity. In *In ACM Multimedia 2005 Proceedings, November 6-11, 2005*, pages 191–198. ACM Press, 2005.

JunctionBox: A Toolkit for Creating Multi-touch Sound Control Interfaces

Lawrence Fyfe
InnoVis Group
University of Calgary
2500 University Drive NW
Calgary, AB T2N 1N4
Canada

Adam Tindale
Alberta College of
Art + Design
1407 14 Avenue NW
Calgary, AB T2N 4R3
Canada

Sheelagh Carpendale
InnoVis Group
University of Calgary
2500 University Drive NW
Calgary, AB T2N 1N4
Canada

ABSTRACT

JunctionBox is a new software toolkit for creating multi-touch interfaces for controlling sound and music. More specifically, the toolkit has special features which make it easy to create TUIO-based touch interfaces for controlling sound engines via Open Sound Control. Programmers using the toolkit have a great deal of freedom to create highly customized interfaces that work on a variety of hardware.

Keywords

Multi-touch, Open Sound Control, Toolkit, TUIO

1. INTRODUCTION

JunctionBox is a new toolkit for building multi-touch interfaces for controlling sound and music that combines existing libraries while adding important new functionality. But why is a new multi-touch toolkit needed and what specifically do sound and music applications require in terms of functionality?

From DIY vision-tracking-based tables to commercially available tablet computers, multi-touch interfaces are becoming a pervasive interaction paradigm. As multi-touch interfaces become increasingly common, it is important for programmers to have high quality toolkits for developing applications that take full advantage of multi-touch hardware. Toolkits can provide the necessary building blocks that help programmers to focus on creative tasks by removing the burdens of low-level implementation, particularly for non-WIMP (window, icon, menu, pointing device) interfaces [2].

One approach to instrument design is to separate interface building from sound engine building (where a sound engine might be Pd [9], ChucK [12], SuperCollider [8] or a similar programmable development environment). In this scenario, information about the state of the interface must be sent to the sound engine. Since the most flexible way to handle messaging is to use Open Sound Control (OSC) [13], any toolkit for developing sound and music control interfaces should have the ability to handle OSC. In addition, a multi-touch sound control toolkit should provide an easy way to map actions on multi-touch hardware to OSC messages.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

Because programmers, including the authors, use a variety of hardware and operating systems, toolkits should, whenever possible, be cross-platform.

The previous points lead to the following basic requirements:

1. Support multi-touch
2. Provide OSC messaging
3. Map multi-touch actions to OSC messages
4. Be cross-platform

Many toolkits exist for building multi-touch applications. The MoMu toolkit [1] maps many input parameters, including touch, on mobile phones (and tablets) to sound control. While MoMu offers a range of sound control possibilities, it is not cross-platform and so cannot be used by programmers who do not choose the hardware and software combination that MoMu requires.

The MT4J toolkit [6] is cross-platform and has multi-touch capability via TUIO [5]. However, it offers no ability to map multi-touch actions to messages. Other toolkits such as PyMT [3] and tuioZones [7] suffer from a similar lack of message mapping capabilities.

2. JUNCTIONBOX

JunctionBox was designed as a toolkit to meet the requirements from Section 1. This section will discuss the incorporated libraries and the classes provided by JunctionBox to programmers. Figure 1 shows the relationship of the incorporated libraries to JunctionBox.

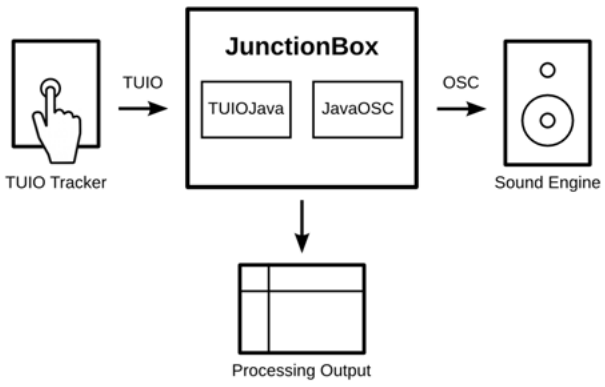


Figure 1: JunctionBox functionality and components.

To make JunctionBox cross-platform, it is written entirely in Java. TUIO was chosen for touch tracking since it has numerous implementations and is decidedly cross-platform, being based on Open Sound Control (OSC). A Java-based TUIO library called TUIOJava [4] provides basic TUIO client functionality. As a TUIO client, JunctionBox can work with any touch tracking systems that meets the TUIO specification. For OSC messaging, a slightly modified version of the JavaOSC [10] library included with TUIOJava is used. For visual output, JunctionBox uses the Processing [11] graphics engine.

The JunctionBox toolkit combines the basic components just described while providing unique functionality via classes described in the following subsections. Figure 2 shows the classes provided by JunctionBox, relating them to external functions like TUIO tracking and OSC messaging.

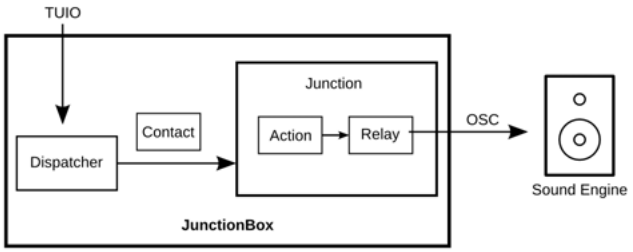


Figure 2: JunctionBox classes shown as boxes.

2.1 Dispatcher

All TUIO message handling in JunctionBox is done via the Dispatcher class. The Dispatcher is a TUIO client but only handles TUIO cursors (touches) and not TUIO objects (fiducial markers). Since not all hardware supports fiducial markers, to be more cross-platform, JunctionBox only handles touch interactions.

The "Box" part of JunctionBox defines the outer limits in width and height of the interactive touch area. This is generally mapped to the size of a touch surface like a video tracking table or a touch tablet. The following line of code creates a new Dispatcher object with a box width and height:

```
Dispatcher d = new Dispatcher(boxWidth, boxHeight,
"127.0.0.1", 6449);
```

The new Dispatcher code contains two additional arguments: a target IP address and port number. These arguments are inherited by Junctions (described below) for sending OSC messages to target sound engines.

2.2 Contacts

When TUIO messages are received by the Dispatcher, TUIO cursors (touches) are converted into Contact objects by the Dispatcher. The Contact class contains the same set of data provided via TUIO 1.1 including session ID, X and Y position, X and Y velocity vectors and acceleration. The Contacts are then dispatched to any Junction whose area coincides with the X,Y position of the TUIO cursor.

2.3 Junctions

The Junction class represents a defined interaction area that offers a set of actions be mapped to messages. Junctions can be created via the Dispatcher:

```
Junction j = d.createJunction(x, y, width,
height);
```

This allows Junctions to inherit values from the Dispatcher like box size and the IP address and port numbers of target sound engines.

A Junction is essentially defined by its area and can take on two shapes: rectangle and ellipse. That area and its location inside of the box determines whether a Junction receives Contacts based on whether touch events occur inside or outside of the area. This is shown in Figure 3.

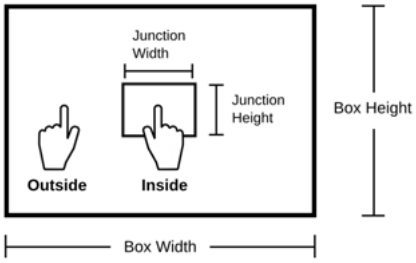


Figure 3: Touches that fall inside or outside of a Junction's area.

Junctions can be rotated, scaled and translated based on the movement of Contacts within a Junction's area. Rotation and translation can be done with a single touch while scaling requires two touches. Each of these actions can be turned on or off at the discretion of the programmer. Additionally, limits can set set on those actions. For example, a limit on Y translation can be coupled with the disabling of X translation to create something like a vertical slider. The following lines of code do this for a Junction j:

```
j.translateX = false;

j.limitTranslateY(100, 500);
```

The last line would limit translation of the Junction to a minimum of Y = 100 and a maximum of Y = 500. Note that these values come from the box size in pixels established when the Dispatcher is created which in turn is related to the canvas size of the Processing sketch containing the visual output code.

Junctions have no inherent visual output but are associated with rectangular and elliptical shapes in Processing. To draw a rectangle in Processing that inherits values from a Junction j:

```
rect(j.getCenterX(), j.getCenterY(), j.getWidth(),
j.getHeight());
```

When the rectangle is drawn in Processing, it will take the current values from the Junction j, so that a translation action will result in the rectangle moving across the screen as the translation occurs. Because Junctions move based on the location of their centers, rectangles and ellipses drawn in Processing must use the center mode to work correctly.

An unlimited number of Junctions can be defined with the Junctions being stackable. When multiple Junctions are created, the last one created receives Contacts where two or more overlap within the box.

A Junction can be added to another Junction. When this is done, the added Junction inherits actions performed on the parent Junction including rotation, scaling and translation.

2.4 Actions

Actions are a means to create mappings between touches and messages for a sound engine. To enable this, the Action

class contains constants whose names correspond to actions built into Junctions.

To add a mapping between an Action and an OSC message requires just one line of code:

```
j.addMessage(Action.TRANSLATE_Y, "/osc/message");
```

Whenever the center Y value of the Junction j changes, that value will be sent as a float argument to "/osc/message". If the center Y value of the Junction j changes to 42, the message will be:

```
/osc/message,f 42
```

For a given Junction, any combination of Actions can be used from none to all. Figure 4 shows some example Actions.

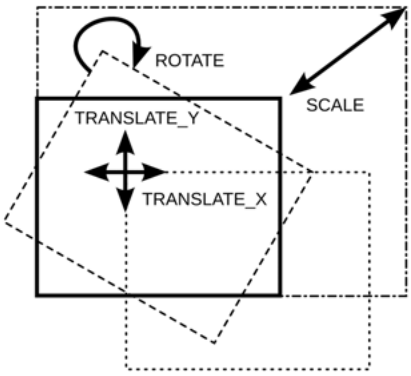


Figure 4: Some Junction actions that can be mapped to messages.

The following actions are available:

- ACTIVE

  If a Junction receives one initial Contact, it will send a message with a single integer value of 1. Messages are only sent when the active state of the Junction changes. So when all contacts have been removed from the Junction, it will send a message with an integer value of 0.

- TOGGLE

  When a Junction receives an initial Contact, it can send a message with its current toggle state. The first Contact sets the toggle state to 1 and triggers a message with that integer value. Any subsequent Contact after the first Contact is removed will trigger a change to the 0 toggle state and that value will be sent in an integer message. This action allows for the creation of simple touch switches.

- ROTATE

  Junction objects can be rotated a full 360 degrees (or less depending on custom limits). Each time the angle of the Junction changes, a float message will be sent out with the current angle normalized from 0 to 1 where 1 represents 360 degrees or 2 Pi radians.

- SCALE

  A two-Contact scaling gesture (where one Contact is held while the other is moved) when applied to a Junction will trigger a calculation of the ratio between the previous area and the current area after scaling. Whenever the scale value changes, a float message is sent with the normalized (0-1) value of the ratio. The normalization works because there is an absolute minimum value for both width and height of a Junction of 1 pixel. The maximum values for width and height of a Junction are the width and height of the box set in the creation of the Dispatcher.

- TRANSLATE_X

  Moving a Junction along the X axis (as defined in Processing) will trigger a float message with the current value of the center X point of the Junction. Messages are only sent when the center moves.

- TRANSLATE_Y

  As with X translation, moving a Junction along the Y axis can trigger a similar float message with the current center Y value of the Junction. Messages are only sent when the center moves.

- TRANSLATE_XY

  Like the above translate actions but with both float values of center X and center Y sent in the same message.

- CONTACT_COUNT

  Whenever the number of Contacts changes, an integer message with the current Contact count is sent.

- ROTATION_COUNT

  Each time a Junction is rotated more than 360 degrees, the current value of the angle is reset to between 0 and 360 degrees. When this is done, a counter for the number of rotations is incremented. This works for clockwise rotations. For counter-clockwise rotations, the angle is negative and the rotation counter is decremented. Any change in the rotation count will trigger an integer message with the current count.

There is no inherent mapping between the chosen actions and the messages sent other than the value associated with the action. While the arguments and their types are fixed, the messages themselves can be changed to any that suit the programmer.

2.5 Relays

The Relay class offers full-featured access to the OSC functionality available in the JavaOSC toolkit with some additional features. Junctions use Relays internally to send messages that are mapped to actions. Outside of Junctions, Relays are designed for occasions when more complicated OSC messaging is required.

A Relay object is created with a target IP address and port number. Then any number of messages can be associated with that target and referenced for later sending.

When using Relays, both the address and the argument number and types can be controlled explicitly. Any action that a Junction can perform can be emulated by getting the current state of Junctions and applying those values directly to messages via Relays.

For example, the following code will create a Relay that sends messages to localhost. Once the message is added to the Relay, the values obtained from a Junction j are added to the message and the message is sent containing the three arguments.

```
Relay r = new Relay(127.0.0.1, 6449);

r.addMessage("/relay/example");
```

```
r.addInteger("/relay/example", j.getToggle());

r.addFloat("/relay/example", j.getAngle());

r.addFloat("/relay/example", j.getCenterX());

r.send("/relay/example");
```

Relays can hold an unlimited number of messages for a given target. Each message can have its arguments set by referencing the message address pattern String as shown in the above example. Additionally, arguments can be added to all messages contained in a Relay with lines like the following that add a float value of 0.5 to all messages.

```
relay.addFloat(0.5);
```

All messages contained in a Relay can be sent by using:

```
relay.send();
```

Also, a list of message strings can be provided to send multiple specific messages.

```
relay.send(messageStrings[]);
```

Using Relays from within Junctions is an easy means of getting multi-touch actions to map to messages. By making the Relay class itself available to programmers, a new set of more complex options becomes available, leaving decisions about messaging and mapping up to the programmer designing the interface without interference from the design of the JunctionBox toolkit.

2.6 Simulator

The Simulator was created for situations where multi-touch hardware is not available and simulates TUIO tracking via the mouse. When used in Processing, the Simulator takes mouse data: whether a mouse button is currently pressed, which button is being pressed, the current X-Y position and the previous X-Y position. That data is then converted to TUIO messages that are received by the Dispatcher object as described above. For now, the Simulator can only simulate a single touch via the mouse.

3. SUMMARY

The JunctionBox toolkit both combines existing libraries for touch tracking and messaging with new features not offered by any existing toolkit. The most significant feature is the ability to easily map multi-touch actions to sound and music control messages.

4. ACKNOWLEDGEMENTS

We would like to thank the Alberta Association of Colleges and Technical Institutes, the Canada Council for the Arts, the Natural Science and Engineering Research Council of Canada, the Alberta Informatics Circle of Research Excellence, SMART Technologies, Alberta Ingenuity, and the Canadian Foundation for Innovation for research support. We would also like to thank the members of the Interactions Lab at the University of Calgary for feedback and support during the development of this project.

5. REFERENCES

[1] N. J. Bryan, J. Herrera, J. Oh, and G. Wang. Momu: A mobile music toolkit. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, pages 174–177, 2010.

[2] S. Greenberg. Promoting creative design through toolkits. In *Proceedings of the Latin-American Conference on Human-Computer Interaction*, CLIHC'09, pages 92–93, November 9-11 2009.

[3] T. E. Hansen, J. P. Hourcade, M. Virbel, S. Patali, and T. Serra. Pymt: a post-wimp multi-touch user interface toolkit. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 17–24, New York, NY, USA, 2009. ACM.

[4] M. Kaltenbrunner. Tuiojava. `http://www.tuio.org/?java`, January 2011.

[5] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. Tuio - a protocol for table-top tangible user interfaces. In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation*, GW 2005, 2005.

[6] U. Laufs, C. Ruff, and J. Zibuschka. Mt4j - a cross-platform multi-touch development. In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '10, New York, NY, USA, 2010. ACM.

[7] J. Lyst. tuiozones. `http://jlyst.com/tz/`, January 2011.

[8] J. McCartney. Rethinking the computer music language: Supercollider. *Computer Music Journal*, 26(4):61–68, 2002.

[9] M. Puckette. Pure data: another integrated computer music environment. In *Proceedings of the International Computer Music Conference*, pages 37–41, 1996.

[10] C. Ramakrishnan. Javaosc. `http://www.illposed.com/software/javaoscdoc/`, January 2011.

[11] C. Reas and B. Fry. Processing: programming for the media arts. *AI & Society*, 20(4):526–538, 2006.

[12] G. Wang and P. Cook. Chuck: a programming language for on-the-fly, real-time audio synthesis and multimedia. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 812–815, New York, NY, USA, 2004. ACM.

[13] M. Wright. Open sound control: an enabling technology for musical networking. *Organised Sound*, 10(3):193–200, 2005.

Beyond Evaluation: Linking Practice and Theory in New Musical Interface Design

Andrew Johnston
Creativity and Cognition Studios
School of Software
University of Technology, Sydney
andrew.johnston@uts.edu.au

ABSTRACT

This paper presents an approach to practice-based research in new musical instrument design. At a high level, the process involves drawing on relevant theories and aesthetic approaches to design new instruments, attempting to identify relevant applied design criteria, and then examining the experiences of performers who use the instruments with particular reference to these criteria. Outcomes of this process include new instruments, theories relating to musician-instrument interaction and a set of design criteria informed by practice and research.

Keywords

practice-based research, evaluation, Human-Computer Interaction, research methods, user studies

1. INTRODUCTION

As its name suggests, the focus of the New Interfaces for Musical Expression (NIME) conference is the development of new musical devices for use in live performance. Thus, a large proportion of NIME papers describe musical interfaces or instruments which show some degree of technical or artistic novelty.

The question of how to evaluate our designs has been a recurring issue. In this paper I present a framework for practice-based research in this area, in the hope that others who pursue similar work will find it of practical benefit. I argue that the process of 'evaluating' new instruments should not be seen as purely an exercise in assessment, but rather as a broader study into performers *and their creative practice* in the context of their use of the new instrument.

1.1 Evaluation and Human-Computer Interaction

Several authors have recognised the potential of human-computer interaction (HCI) techniques to investigate the experiences of performers who use musical interfaces. In general, the approach has been to use quantitative techniques from HCI which tend to equate interface effectiveness with efficiency. Wanderley and Orio [15], for example, propose a series of "musical tasks" which might be used in order to evaluate how effectively an input device can support expressive performance. These tasks are intended to create a kind of benchmark which will make it easier to compare one interface device with another. The intention is that these benchmark figures, derived as they are from formal studies of users doing prescribed musical tasks, might complement traditional technical measures of device capabilities such as output rate and precision.

This is certainly worthwhile. However, this approach is very much focussed on the devices and their ability to efficiently translate the intentions of the user into parameters for the computer. The experiences of the users who use the devices, being harder to quantify, are comparatively neglected.

To address this, we need to broaden the scope of what constitutes 'evaluation' in this context, and acknowledge that while ergonomics and efficiency are important, they are not the primary determinants of the quality of a musical interface. This thinking is reflected in the broader field of HCI, where there has been recognition that the task-based approach alone is inadequate, particularly when considering software intended to support creative work. A number of HCI researchers therefore have turned their attention to the 'user experience' [1, 10].

In addition, some researchers are proposing new ways of thinking about 'evaluation' in the context of systems which have uses that are open to a range of interpretations. Sengers and Gaver [14], for example, argue that interaction designers are becoming less concerned with designing software which unambiguously conveys and supports a clearly defined 'purpose'. They propose that HCI needs to support interactions in which users may have multiple interpretations of what a system is for and how it works. 'Evaluation' in this context goes beyond identifying whether users' interpretations of a system's purpose and behaviour matches the designer's anticipated interpretation. Rather,

> "evaluation shifts from determining whether an authoritative interpretation was successfully communicated to identifying, coordinating, stimulating, and analyzing processes of (evaluative) interpretation in practice" [14], p. 105

This approach suggests we move beyond 'evaluating' our interface designs, and use examination of users' experiences to support reflection on both musical interface design and the nature of the activities they afford. That is, we move beyond evaluating how effective our designs are at supporting musical expression and instead use them as provocative prototypes [12] which stimulate examination of the nature of expression itself – at least as it occurs in a particular cultural context.

Given this significant broadening of scope, it is timely to consider whether the term 'evaluation' is still appropriate. In my view, evaluation is best seen as a *component* of a broader examination of both musical interface design and

NIME'11, 30 May–1 June 2011, Oslo, Norway.

musical expression and I therefore argue that a more general term such as 'user experience study' is preferable. It is certainly important that we evaluate our instruments - that we assess how well they meet relevant criteria - but because our design criteria embody our theories of designing for musical expression, we should be equally interested in refining, or redefining, the criteria.

2. RESEARCH STRUCTURE

The research process I have adopted is shown in figure 1. Initial design criteria, drawn from the literature and personal experience inform the design of new musical interfaces (or instruments for want of a better term). From the design process we get the musical interfaces themselves and a set of design criteria which the designer believes they embody. These instruments, and the experiences of musicians who use them, are scrutinised in a series of user studies. From these studies we gain theoretical understanding of musical performance with instruments of this kind and a refined set of design criteria informed by practice and research.

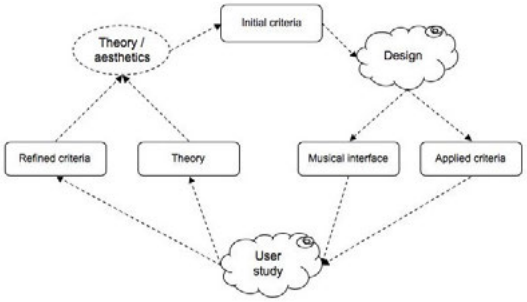


Figure 1: Overview of the research process.

I will consider this process in some detail in the following sub-sections.

3. DESIGN CRITERIA

When presenting new musical instruments, it is possible to describe the instruments in purely functional terms - to outline exactly what they do - but it is equally important to consider *why* they work in this way. Of course, the instruments behave the way they do because the designers made them that way, with particular goals in mind. Identifying these goals - the design criteria - helps make the intentions of the designers explicit and therefore open to examination and discussion. It also facilitates evaluation of the new instruments in terms of the criteria.

Portillo and Dohr define a design criterion as, "a measure of value used by the designer to conceptualize, test and evaluate the project purpose in the design process" [13], p.405. An important point to note here is that design criteria are used, perhaps tacitly, during the design process as the designers develop their ideas and the designed artefacts in parallel. Making these criteria explicit may not be easy, and producing an exhaustively complete list of all criteria that were applied is impractical and probably not particularly helpful. A balance and appropriate level of granularity needs to be found.

In my research I have drawn on several sources to identify initial sets of design criteria that were most influential in shaping the instruments I have created. These include:

- Reflective online diaries (blogs) kept by those involved in the design process (composers, musicians, software developers, etc).
- Interviews with artists and designers involved in the development of the instruments.
- Examination of software version control logs.[1]

It is important to note that it is not expected that all the applied design criteria should, or could, be identified at this stage. Criteria identified prior to the user studies help focus the interviews and user studies which follow. We expect, and hope, that criteria will be significantly refined and added to as the research progresses.

4. USER EXPERIENCE STUDIES

User studies are primarily concerned with three questions:

1. Do the instruments that have been created meet the design criteria identified during design?
2. How do musicians experience them?
3. What are the relationships between the characteristics of the instruments and the musicians' experiences?

I have conducted studies of a series of instruments [6, 8, 9], and in this section I briefly outline the approach to data gathering and analysis. Data was gathered from the following sources:

- A user study in which seven professional musicians were videoed playing with the instruments, commenting on their experiences and responding to interview questions. This was the primary source of data.
- Notes made by observers who attended the musicians' sessions with the instruments. These provided additional perspective on the musicians' experiences and helped identify whether the instruments met the design criteria.
- Questionnaires administered during the musicians' sessions with the instruments which attempted to directly elicit their opinion on whether it met the design criteria.

The question of whether the instruments met the design criteria was primarily addressed by analysing the questionnaires and the notes made by the observer. The more complex question of how musicians interacted with the instruments was the primary focus of the study. In order to address this question, the video recordings of the musicians' sessions with the virtual instruments were transcribed and the grounded theory method was used to generate a theory of musician - interface interaction. The observers' notes provided additional perspective on this data and informed the development of the theory.

The studies involved seven professional musicians who had a minimum of five years professional experience. They included principal players from professional symphony orchestras as well as leading improvisers. Due to the degree of expertise of the participants and the in-depth nature of the evaluation, this was a sufficient number to provide detailed insight into the experiences of expert musicians with the virtual instruments. Note that in qualitative research

[1]Version control software (eg. Subversion) is used during software development to track changes at every stage of the design process.

the emphasis is on *generating*, rather than validating, theory [5]. As such, this research was intended to provide detailed insight into the experiences of the specific musicians who participated in the evaluation, and to generate theories consistent with what was observed. It is hoped that this research will provide a sound basis for future research which may attempt to more broadly *validate* the concepts and relationships uncovered in this study. Such validation would be likely to involve larger numbers of musicians using virtual instruments in a simplified and more controlled context.

The focus of the investigation was on what the musicians were able to do with the virtual instruments, what impact using them had on their music making and any suggestions for improvements, so the musicians were not asked to perform specific musical tasks. Rather, they were told in simple terms how the virtual instruments behaved and then asked to explore and make music with them.

The musicians were asked to verbally reflect on their experience with the instruments using a variation of the 'think aloud' approach [3]. When using the concurrent think-aloud approach, the idea is that the musicians continuously verbalise what is going through their mind as they use the instruments, keeping the time between thought and verbal expression to a minimum. However, asking musicians to generate fully concurrent think-aloud reports presented obvious practical problems because wind and brass musicians are unable to speak (intelligibly) and play their instrument at the same time. A sensible compromise was to ask the musicians to verbally report what they were thinking and perceiving as frequently as they were able during their time using the instruments. This meant that they were effectively providing a large number of smaller retrospective reports as they played for a time, commented on what was happening, played some more, made further comments and so on.

In addition to gathering information about what the musician was thinking and experiencing as they used the virtual instruments, the musician's opinions on the instruments and suggestions for how they could be improved were actively solicited. As experts in their field, it was hoped that the musicians would be able to provide insight into the nature of the virtual instruments, their potential uses, limitations and areas for improvement. The intention was that the musicians would become engaged with the design process and in a sense become co-designers. As such, the format of the evaluation was flexible. There was a standard procedure but when interesting issues arose, this was varied. Because the emphasis of this study was on theory generation rather than verification, the gathering of rich data was prioritised over consistency of procedure. The process was more akin to a user dialogue than usability testing [2].

After using each virtual instrument, a semi-structured interview was conducted in which participants were asked a series of open questions relating to their experience with the virtual instrument. In order to facilitate later analysis, the musicians' interaction with the instruments and the interviews were video recorded.

4.1 Data Analysis

The video-recordings of the musicians playing the virtual instruments and talking about their experiences were a very rich source of data. A challenge was to identify consistent themes and patterns in order to make sense of this information. Techniques from the grounded theory method [5, 4] were therefore used to code and analyse the data gathered. This method was a good fit for this purpose because it facilitated the generation of theory closely tied to the evidence from rich qualitative data. At a high level, the basic steps of the grounded theory analysis process as applied in this study were:

- Transcribing the evaluation sessions.
- Open coding: that is, identifying and labelling incidents in the data (including non-verbal data). This is done line by line, coding each sentence. As coding progresses, incidents are constantly compared with one another to identify similarities and differences.
- Memoing: as ideas emerge regarding the codes and their relationships during coding, the researcher stops to make a note. Memoing aids the process of linking the descriptive codes into theory.
- Sorting: memos are sorted and arranged in order to identify core issues and their relationships with one another and thus build theory which is 'grounded' in the gathered qualitative data.

In my work I have made use of the open-source software Transana [16] to facilitate this process. With Transana, clips of interesting video data can be created and labelled with codes (known as 'keywords in Transana) which are specified by the researcher. Once coding is complete, searches can be made which find all clips assigned particular codes. For example, a search could be made which found all video clips from all participants which were assigned the code 'control'. Each of these clips could be examined in detail to find key points of similarity and difference. These features were invaluable when dealing with the more than fourteen hours of video gathered during the studies.

4.2 Building Theory

Obviously, merely labelling incidents in the data does not create theory, but building up a coding scheme in this way facilitates what Glaser and Strauss [5] describe as the 'constant comparison' technique. Constant comparison simply involves comparing incidents in the data with one another, identifying similarities, differences and relationships which are recorded in memos as the researcher identifies them. In the grounded theory method, memoing is the process by which the analyst reflects upon and documents their evolving understanding of the situation under study. Memoing also helps the analyst to link the codes together into a theoretical framework. Memos are simply notes written by the researcher. They do not have a required format, the intention being simply that insights are captured quickly so that they are retained. Memoing in this study made use of a feature of Transana which allows the researcher to attach 'notes' to transcripts or collections of clips.

Through this process the researcher builds a theory which helps to make sense of the situation under examination. Memos help facilitate and, to some extent, document the researcher's evolving understanding of the links between these incidents. However, it is important to note that memos and coding schemes are not a complete record of the analysis process. In my research, coding and memo-writing are undertaken primarily to facilitate analysis rather than document it. Thus the coding scheme and memos should be considered a by-product of the analysis process which generates theory.

The fundamental idea is that the researcher examines the codes that have been created during open coding and attempts to identify higher-level concepts that make apparent patterns in the codes, and relationships between them. The approach described above draws primarily on the suggestions of Glaser [4] and Miles and Huberman [11].

4.3 Findings

This paper is primarily concerned with research methods and space precludes a detailed discussion of findings which have been published elsewhere [6, 7, 8]. However, I will briefly outline some key findings in order to illustrate the kinds of conclusions that can be drawn from a study of this type.

During grounded theory analysis it is expected that open coding will lead to the discovery of a 'core' category, a key issue which appears to have particular relevance to the situation under study [5]. The core category emerges during analysis as the researcher continually compares incidents in the data, noting relationships between incidents in memos.

When analysing the data gathered during this study, it was clear that the musicians did not always approach the virtual instruments in the same way. Sometimes a musician would express frustration because they felt they did not have enough control over the virtual instruments, but then at other times the same musician would complain that the virtual instruments were not autonomous enough, and that they wanted their behaviour to be less predictable. It seemed that the qualities the musicians sought in a virtual instrument would change during their interactions - that they interacted with the virtual instruments in different modes. Thus the core issue which emerged during analysis was that of *modes of interaction*.

We found that the musicians interactions with the virtual instruments could be classified into three modes: instrumental, ornamental and conversational. In instrumental mode the musician seeks a high level of detailed control over all aspects of the virtual instrument's behaviour. Musicians taking an instrumental approach essentially see the virtual instrument as an extension of their acoustic instrument and want it to respond consistently so that they can trust it during performances.

In ornamental mode, musicians surrender detailed control of the generated sound and visuals and let the virtual instrument create audio-visual layers that are added to their acoustic sounds. Musicians taking an ornamental approach may not pay active attention to the behaviour of the virtual instrument, instead leaving it to its own devices and expecting (or hoping) that it will do something that complements or augments their sound without requiring directed manipulation.

Conversational interaction occurs when musician approaches the virtual instrument as a musical partner. In conversational interaction the musician allows the virtual instrument to 'talk back', at times directly influencing the overall direction of the music. The musical 'balance of power' is in flux as responsibility for shaping musical direction continually shifts between musician and virtual instrument.

5. CONCLUSION

In this paper I have detailed an approach to linking practice and theory in musical interface design. The guiding principles of this method have been described and I have summarised how it has been applied to generate and refine theory concerning the nature of performers' interactions with musical interfaces.

The outcomes of the practice-based research process I have outlined are a set of musical interfaces, a theory of musician-instrument interaction and a set of design criteria informed by practice and research.

I believe that criteria-based evaluation and qualitative user studies are a simple, yet powerful combination which enables a form of detailed and rigourous reflection on the creative outcomes of musical interface design. The specific methodological choices I have made in relation to how to gather and analyse data were driven by the particular characteristics of the musical interfaces we designed and the aesthetic goals which guided their development. Thus, I do not propose this method as a detailed one-size-fits-all solution, but hope that discussion of this work will encourage a broader view of 'evaluation' in musical interface design and help practitioners and researchers more effectively link practice and theory.

6. REFERENCES

[1] M. A. Blythe, K. Overbeeke, A. F. Monk, and P. C. Wright. *Funology: from usability to enjoyment.* Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[2] J. Buur and K. Bagger. Replacing usability testing with user dialogue. *Communications of the ACM*, 42(5):63–66, 1999.

[3] K. A. Ericsson and H. A. Simon. *Protocol Analysis: Verbal Reports as Data.* MIT Press, Cambridge, MA, revised edition, 1993.

[4] B. G. Glaser. *Theoretical Sensitivity.* The Sociology Press, 1978.

[5] B. G. Glaser and A. L. Strauss. *The discovery of grounded theory: strategies for qualitative research.* Aldine de Gruyter, New York, 1967.

[6] A. Johnston. *Interfaces for Musical Expression Based on Simulated Physical Models.* PhD thesis, University of Technology Sydney, 2009.

[7] A. Johnston, L. Candy, and E. Edmonds. Designing and evaluating virtual musical instruments: facilitating conversational user interaction. *Design Studies*, 29(6):556–571, 2008.

[8] A. Johnston, L. Candy, and E. Edmonds. Designing for conversational interaction. In *Proceedings of New Interfaces for Musical Expression (NIME)*, 2009.

[9] A. Johnston, B. Marks, and L. Candy. Sound controlled musical instruments based on physical models. In *Proceedings of the 2007 International Computer Music Conference*, pages vol1: 232–239, 2007.

[10] J. McCarthy and P. Wright. *Technology as Experience.* The MIT Press, 2007.

[11] M. B. Miles and A. M. Huberman. *Qualitative Data Analysis: An Expanded Sourcebook(2nd Edition).* Sage Publications, Inc, 1994.

[12] P. Mogensen. Towards a provotyping approach in systems development. *Scandinavian Journal of Information Systems*, 4:31–53, 1992.

[13] M. Portillo and J. H. Dohr. Bridging process and structure through criteria. *Design Studies*, 15(4):403–416, 1994.

[14] P. Sengers and B. Gaver. Staying open to interpretation: engaging multiple meanings in design and evaluation. In *DIS '06: Proceedings of the 6th conference on Designing Interactive systems*, pages 99–108, New York, NY, USA, 2006. ACM.

[15] M. M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from HCI. *Computer Music Journal*, 26(3):62–76, 2002.

[16] D. Woods and C. Fassnacht. Transana v2.22. Madison, WI: The Board of Regents of the University of Wisconsin, 2007.

Intuitive Real-Time Control of Spectral Model Synthesis

Phillip Popp
Oakland, CA 94612
popp.phillip@gmail.com

Matthew Wright
CREATE/MAT
University of California
Santa Barbara, CA 93106
matt@create.ucsb.edu

ABSTRACT

Several methods exist for manipulating spectral models either by applying transformations via higher level features or by providing in-depth offline editing capabilities. In contrast, our system aims for direct, full, intuitive, real-time control without exposing any spectral model features to the user. The system extends upon previous machine learning work in gesture-synthesis mapping by applying it to spectral models; these are a unique and interesting use case in that they are capable of reproducing real world recordings, due to their relatively high data rate and complex, intertwined and synergetic structure. To achieve a direct and intuitive control of a spectral model, a method to extract an individualized mapping between Wacom Pen parameters and Spectral Model Synthesis frames is described and implemented as a standalone application. The method works by capturing tablet parameters as the user pantomimes to synthesized spectral model. A transformation from Wacom Pen parameters to gestures is obtained by extracting features from the pen and then transforming those features using Principal Component Analysis. Then a linear model maps between gestures and higher level features of the spectral model frames while a k-nearest neighbor algorithm maps between gestures and normalized spectral model frames.

Keywords

Spectral Model Synthesis, Gesture Recognition, Synthesis Control, Wacom Tablet, Machine Learning

1. INTRODUCTION

Spectral Model Synthesis (SMS) is a flexible platform capable of generating rich and vivid sounds [11] [9]. It represents a sound's periodic and noisy components as a series of frames, each frame consisting of a set of sinusoidal frequencies and amplitudes plus a spectral envelope for noise. Deriving a compact spectral model from recorded audio captures a veridicality difficult to create using other forms of synthesis. SMS retains the gestalt of the audio while allowing stretching, pitch shifting and other modifications. Despite this flexibility, SMS is difficult to manipulate intuitively in real-time beyond macro control such as volume, pitch, and duration. The number of synthesis parameters in a single SMS frame can be well over 100; choosing how to tie a low-dimensional control device to these is non-trivial.

Recently, machine learning and statistical analysis techniques have been applied to mapping inputs to synthesizer controls [2]. We apply these ideas to SMS and propose a new model to map input gestures to SMS control. We tailor each mapping in a user-directed way by having the user listen as a spectral model is resynthesized and pantomime, in real-time, the gestures that "should" correspond to the sound. Essentially, the user imagines that she is directly controlling the sound with a Wacom Tablet [12]. The system captures these pantomimed input gestures for use as a training set to determine a mapping between the tablet and SMS parameters via machine learning techniques.

The first learning step analyzes the captured input gestures with principal component analysis (PCA) to create a lower dimensional "gesture" space. Linear regression then maps between the (principal components of the) gestures and higher level spectral model frame features, while k-nearest neighbors maps between the gestures and normalized spectral model frames. After the system learns a complete mapping it can synthesize new sounds in response to real-time Wacom gestures. Since the mapping originated from the user's pantomimes to the original spectral model, the control is intuitive. Repeating the example gestures results in approximating the original SMS frames, while deviating from the original pantomimes results in new spectral model frames that did not exist in the original spectral model but make intuitive perceptual sense to the user.

2. RELATED WORKS

There have been several approaches to controlling SMS. One approach focuses on providing software tools to allow users to edit spectral models in an offline manner [5]. A second approach reduces the number of inputs needed to control synthesis by extracting higher level sonic descriptors derived from the spectral model [8], using general purpose dimension reduction techniques [6], or defining generic *a priori* mappings [13]. Instead, our approach allows users to map their personal gestures to aspects of the spectral model rather than simply mapping the model characteristics to a parameter value. Like the second approach, it attempts to control the synthesis in a higher-level and more abstract space, but in contrast it provides a personalizable and potentially more flexible platform because each user can reconfigure the control mapping for each spectral model.

Fiebrink et al. utilize various machine learning and signal processing algorithms to map between input controllers and synthesis controls [2], demonstrating this approach by applying it to score following, physical modeling synthesis, and video manipulation utilizing a "play-along" data gathering method. Our approach differs in both the synthesizer's structure and the aspirations of the control mapping. Particularly, SMS provides a more low-level representation of sound than musical scores or physical models. The data rate

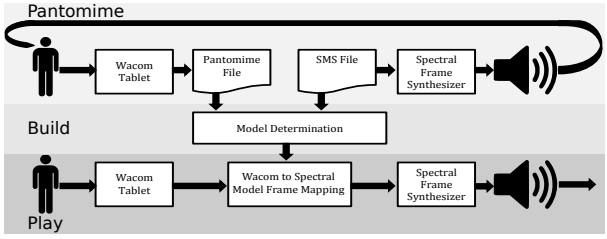


Figure 1: Overview of Mapping Generation Steps

and emergent nature of the information in SMS culminate in rich, detailed and life-like sounds while making intuitive control of the model exponentially more difficult. As well as SMS structurally differing from the previously investigated synthesis models, the source and derivation of the model is also conceptually different. Fiebrink et al. utilize scores and/or random permutations of synthesis settings to provide examples for users to pantomime to. Here, the model is derived from a recording, and since spectral models are so flexible, special care must be taken to retain the gestalt of the sound while still offering new spaces for exploration. The spectral model synthesizer in itself is cannot retain the essence of the recording. Instead it is the duty of the the machine learning algorithms to retain certain qualities of the model, while relaxing others in order to provide direct, intuitive control while still being capable of creating unforeseen sounds. To do so we utilize expert knowledge of SMS, pervious work upon motion gesture analysis [1] and machine learning for synthesis control [2], and provide an environment where users can experiment to create new mappings [7].

3. SYSTEM OVERVIEW

Our method uses several steps to learn a mapping from tablet input to SMS control (figure 1). First the user generates examples by pantomiming "control" gestures as a predetermined model plays; the system time-stamps and records the resulting tablet parameters into a *pantomime file*. Then the machine learning engine derives a two-level mapping: from pen parameters to *gestures*, and from these gestures to SMS frames. One can then use this mapping in real-time to control SMS using the tablet. A standalone OSX application guides the user through the entire process, from pantomiming, to building a mapping, to playing.

3.1 Pantomimes

Our software lets the user load a spectral model, preview it, pantomime to it, and see the Wacom Pen's parameters. An example of a pen's parameters in comparison to a spectral model's frequencies can be seen in figure 2. The software provides three mechanisms to aid the user. The first is visual feedback of the recent history of all pen parameters shown as a trail of slowly fading dots upon a white canvas: X and Y position determine dot's position, pen-tip pressure and Z position (pen's height above tablet) determine hue, and the X and Y tilt parameters control the size and shape of the dot. Second, the user can learn the nuances of a chosen spectral model via practice runs listening and pantomiming without recording the results. Third, to help accurately synchronize the pantomime timing to the sound, a stop-light metaphor counts down (via large red, then yellow, then green circles each displayed for one second) to the beginning of audio playback after the user clicks the record button. This also gives the user time to prepare (e.g., picking up the Wacom pen after clicking the record button).

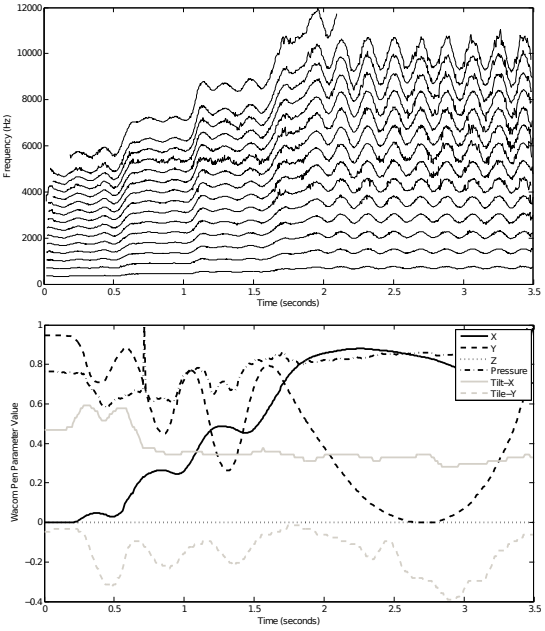


Figure 2: Sinusoidal Tracks of Spectral Model (top) and Wacom Pen Parameters (bottom)

3.2 Gesture Language Extraction and Transformation

Given the opportunity to pantomime to SMS resynthesis, each user will assuredly perform different gestures to the same audio. Likewise, the same user will usually perform widely differing gestures when pantomiming to different SMS models [3]. These gestures should reflect the way a user would intuitively control the sound if they were producing it. In order to create individualized and intuitive control of SMS parameters, features likely to express musical intention are extracted from the captured Wacom Tablet parameters. We assume that features containing relatively high energy amongst the set of captured input device parameters encapsulate a high expressive potential, according to the principle of a "correspondence between the "size" of a control gesture and the acoustic result" [10]. To this end, we define our gesture language as several linear combinations of features with high expressive potential and derive a transformation between features and the gesture language using Principal Component Analysis (PCA). This transformation emphasizes features with high expressive potential and deemphasizes those with lower expressive potential, as well as reducing the dimensionality of the input to the learning algorithms in later stages of this mapping algorithm.

3.2.1 Feature Extraction

To extend our mapping algorithm's ability to encapsulate gestures, we estimate the instantaneous velocity (first derivative) and acceleration (the second derivative) of the tablet parameters using a five-point stencil. This preprocessing step is primarily to capture non-linear motion information. We define the set of six parameters (x, y, and z position, pressure, x-tilt, y-tilt) from the Wacom Pen as follows:

$$\mathbf{p}(n) = [w_x(n), w_y(n), w_z(n), w_p(n), w_\theta(n), w_\phi(n)]. \quad (1)$$

Each output frame is the concatenation of the original pen parameters with the first and second derivatives:

$$\mathbf{f}(n) = [w_x(n), w'_x(n), w''_x(n), ..., w_\phi(n), w'_\phi(n), w''_\phi(n)]^T \quad (2)$$

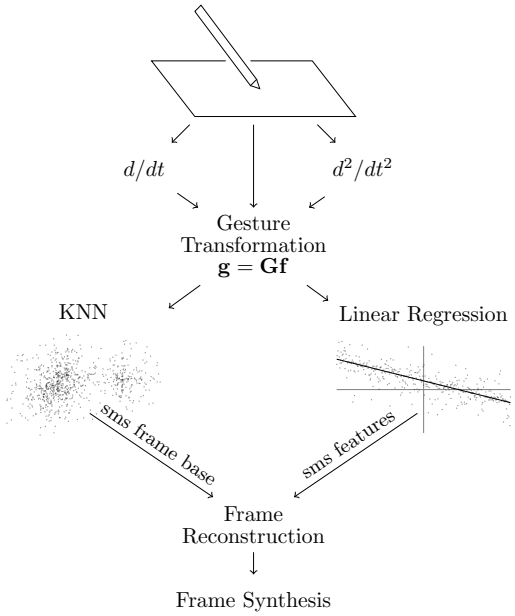


Figure 3: Mapping From a Wacom Tablet to Spectral Model Frames

3.2.2 Gesture Language

We aim to determine a gesture language that distinguishes and accentuates the features that show high potential expressivity for a particular user, as well as a means to transform the tablet features into the gesture language in a real-time fashion. Principal Component Analysis (PCA) provides a suitable means to determine a transformation to a gesture language of M continuous values (where $0 < M \leq 18$). If we associate energy with expressivity, PCA results in a transformation where the first column of the transformation captures the most expressivity, the second column contains the second most expressivity, the third column the third most expressivity, and so on. Additionally, we emphasize/deemphasize the output of the transformation by weighting each of the M columns by their respective eigenvalues. In addition to accentuating Wacom Pen features used for expression, the gesture transformation also reduces the dimensionality of inputs to the next mapping stage, making them more robust against the various pitfalls associated with the curse of dimensionality [4]. Equation 4 describes the transformation matrix between Wacom Pen features and gestures where λ_i is the eigenvalue and $\mathbf{p}_i$ is the corresponding eigenvector derived from PCA analysis of $\mathbf{F}$. The transformation from Wacom Pen parameters to gestures $\mathbf{g}(n)$ is shown by equations 1, 2 and 5.

$$\mathbf{F} = [\mathbf{f}(0), \mathbf{f}(1), ..., \mathbf{f}(N-1)]^T \tag{3}$$

$$\mathbf{G} = [\lambda_0 \mathbf{p}_0, \lambda_1 \mathbf{p}_1, ...\lambda_{M-1} \mathbf{p}_{M-1}]^T \tag{4}$$

$$\mathbf{g}(n) = \mathbf{G}\mathbf{f}(n) \tag{5}$$

3.3 Gestures to Spectral Model Frames

It would be challenging to find a one-size-fits-all mapping from gestures to spectral model frames. Spectral model frames posses a composite structure, made of components that have disparate meanings and values. To overcome this challenge we employ the two-pronged approach outlined in figure 3. One mapping path maps gestures to higher level spectral frame features via linear regression. The other path utilizes a K-Nearest Neighbor (KNN) algorithm where the gesture values describe the coordinates of normalized spectral model frames in a Euclidean space. After the linear regression and KNN models are trained, a new spectral model frame is generated by advancing a gesture through both mapping paths and then combining their output to construct a new spectral model frame. Retaining the normalized frames preserves the complex structure of the spectral model, which in turn contains many of the minute details that differentiate SMS from other forms of synthesis. Concurrently, linearly mapping higher level features allows new sonic spaces to be explored.

3.3.1 Linear Mapping of Higher Level Features

Linearly mapping higher level features expands the capability of the overall mapping algorithm by encapsulating complex features of the spectral model and providing unbounded control of them. Higher level features can capture aspects of the spectral model that happen on a time-scale too small to recreate accurately by drawing, or in a way that maps complexly to gestures. Consider mapping a spectral model's vibrato. The pitch fluctuations happen on a time scale too small to reproduce when pantomiming. By instead mapping a gesture to the depth of a vibrato, the user simply needs to pantomime something that implies more vibrato, not match the fluctuations in pitch directly. Linear mapping also allows the system to generalize parameters beyond the range presented in the original spectral model. For example, if we only used a KNN approach to map pitch, the user would be confined to the pitches existent in the original spectral model. By deriving a linear mapping, the user can go beyond and between the original pitches because the linear map utilizes an unbounded continuous function to convert gestures to pitch.

To create a linear mapping of a higher level feature, first the feature is extracted from the spectral model frames. The function used to extract the feature must normalize the spectral model frame with respect to the feature as well as be invertible so that the spectral model frame could be recreated at a later stage. These higher level features are then mapped linearly by performing linear regression upon the pairs of gestures and their corresponding feature value.

3.3.2 KNN Mapping of Spectral Model Frames

After extracting higher level features from the spectral model frames, the normalized spectral model frames are placed in a Euclidean space where the values of the gesture are utilized as the frame's coordinates.

A KNN algorithm is used to map between gestures and these altered spectral model frames. KNN algorithms work on the assumption that data can be arranged into a metric space, and that a new, unclassified piece of data can best be described by inspecting the K nearest classified data within a training set [4]. This property is extremely attractive in that it allows us to use our gesture vectors as direct predictors of the output spectral model frame, ignoring the complex relationship between the gesture vector and the particular format of the SMS frame.

3.3.3 Spectral Model Frame Reconstruction

After an input gesture has been mapped through the higher level feature linear models and the normalized spectral model frame KNN model, a new spectral model frame is constructed by combining the two. Beginning with the base spectral model frame at the output of the KNN, the higher level features are applied to the frame using the inverse of the function used to extract the higher level feature. This frame is then fed to the synthesizer for audio playback.

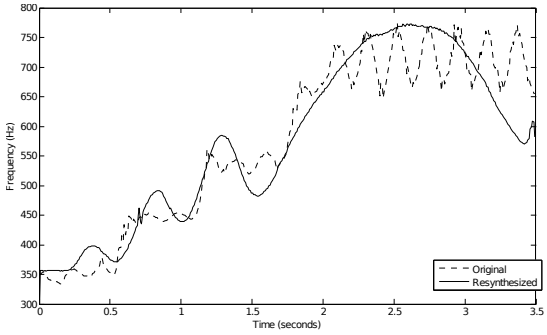


Figure 4: Original and Mapped Fundamental Frequency

4. RESULTS/ANALYSIS

The system as a whole successfully allows intuitive control over a spectral model. It could deduce a mapping at a very satisfactory speed, quick enough for fast experimentation. For a spectral model covering 4 seconds of time, and a pantomime file containing 428 frames it took approximately 2.5 seconds to derive a mapping using a single Intel Core i5 on a Macbook. While the speed of the system does not pose an issue, certain aspects of the mapping algorithm do. First, the user has the option to pantomime several times to the same spectral model and combine all of their pantomime files into a single training run. It was noted that a single pantomime generally produces satisfactory results, but additional pantomime files smoothed out the control over the spectral model, and made it feel more predictable. We hypothesize that this is simply a matter of providing more training data, resulting in more robust calculations in both the PCA and linear regression stages. Additional investigation is needed to shed more light on the appropriate amount of training data needed to derive a control mapping. Also, while many aspects of the original spectral model (loudness, frequency envelope) could be recreated by reproducing the original pantomimes, the resynthesized sound lacked the same authenticity in the original spectral model. Figure 4 shows both the spectral model's original fundamental frequency, and the fundamental frequency determined by mapping the pantomime file through the mapping algorithm. While the general trends of the original fundamental are grossly estimated, many of the finer temporal variations are completely smoothed out. One possible explanation could be that the Wacom Pen's sample rate is too slow (∼50Hz) and control too gross. This makes it incapable of controlling micro-variations of spectral model parameters that change more often than the Wacom Pen is sampled. Second, the linear regression between gestures and the fundamental frequency may be too simple of a model to translate from gestures to the fundamental. Additional investigation is needed to understand exactly which features of the spectral model are retained after the mapping, and which are lost.

5. CONCLUSION AND FUTURE WORK

We have introduced a novel method to extract an intuitive and individualized mapping from a users's performance to control of Spectral Model Synthesis based on capturing tablet parameters as the user pantomimes to synthesized spectral model. A robust machine learning system incorporating time derivative estimation, PCA, KNN, and linear regression produces acceptable results: when the performance gestures imitate the pantomimed training gestures the output sound recognizably approximates the input sound, while related gestures intuitively produce interesting extrapolated sounds.

Several areas of improvement could increase the overall quality of the system. First, a better methodology for pairing Wacom Pen parameters to spectral model frames could improve the overall mapping by reducing the inherent errors of a pantomime. By segmenting both the Wacom Pen parameters and spectral model into sub-note sections (e.g. attack, decay, sustain), we could come to a tightly bound pairing between gestures and sub-note events. Second, additional mapping techniques could be investigated such as artificial neural networks, and logit regression. While these models may require more training and time to create, they have the capability to capture complex inter-feature relationships not captured by the linear regression functions of the current design.

6. REFERENCES

[1] F. Bevilacqua, J. Ridenour, and D. Cuccia. 3D motion capture data: motion analysis and mapping to music. In *Proceedings of the workshop/symposium on sensing and input for media-centric systems*. Citeseer, 2002.

[2] R. Fiebrink, P. Cook, and D. Trueman. Play-along mapping of musical controllers. In *Proc. International Computer Music Conference*. Citeseer, 2009.

[3] R. Godøy, E. Haga, and A. Jensenius. Playing air instruments: Mimicry of sound-producing gestures by novices and experts. *Gesture in Human-Computer Interaction and Simulation*, pages 256–267, 2006.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. Springer, 2009.

[5] M. Klingbeil. Software for spectral analysis, editing, and synthesis. In *Proceedings of the International Computer Music Conference*, pages 107–110. Citeseer, 2005.

[6] S. Le Groux and P. Verschure. Perceptsynth: mapping perceptual musical features to sound synthesis parameters. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 125–128. IEEE, 2008.

[7] P. Popp. `http://www.phillippopp.com`.

[8] X. Serra and J. Bonada. Sound transformations based on the sms high level attributes. In *Proceedings of the Digital Audio Effects Workshop*. Citeseer, 1998.

[9] X. Serra and J. Smith. A sound decomposition system based on a deterministic plus residual model. *The Journal of the Acoustical Society of America*, 87:S97, 1990.

[10] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

[11] M. Wright, J. Beauchamp, K. Fitz, X. Rodet, A. Röbel, X. Serra, and G. Wakefield. Analysis/synthesis comparison. *Organised Sound*, 5:173–189, 2000/12/01 2000.

[12] M. Wright, D. Wessel, and A. Freed. New musical control structures from standard gestural controllers. In *Proceedings of the ICMC*, 1997.

[13] M. Zbyszynski, M. Wright, A. Momeni, and D. Cullen. Ten years of tablet musical interfaces at CNMAT. In *Proceedings of the 7th international conference on New interfaces for Musical Expression*, pages 100–105. ACM, 2007.

BeatJockey: A new tool for enhancing DJ skills

Pablo Molina*, Martín Haro**, Sergi Jordá**
Music Technology Group
Universitat Pompeu Fabra
Roc Boronat, 138, 08018 Barcelona, Spain
*faival@gmail.com, **name.surname@upf.edu

ABSTRACT

We present *BeatJockey*, a prototype interface which makes use of Audio Mosaicing (AM), beat-tracking and machine learning techniques, for supporting Diskjockeys (DJs) by proposing them new ways of interaction with the songs on the DJ's playlist. This prototype introduces a new paradigm to DJing in which the user has the capability to mix songs interacting with beat-units that accompany the DJ's mix. For this type of interaction, the system suggests song slices taken from songs selected from a playlist, which could go well with the beats of whatever master song is being played. In addition the system allows the synchronization of multiple songs, thus permitting flexible, coherent and rapid progressions in the DJ's mix. *BeatJockey* uses the Reactable, a musical tangible user interface (TUI), and it has been designed to be used by all DJs regardless of their level of expertise, as the system helps the novice while bringing new creative opportunities to the expert.

Keywords

DJ, music information retrieval, audio mosaicing, percussion, turntable, beat-mash, interactive music interfaces, real-time, tabletop interaction, reactable.

1. INTRODUCTION

After the term Diskjockey (DJ) was coined in the early 30s, the first DJs used a single device to sequentially playback songs on the radio (Radio-DJ). Afterwards, the appearance of portable turntables on the scene inspired club/rave DJs to use two turntables and a mixer. In Jamaica, Scratching-DJs [9], and Mixing-DJs [25], started to increase complex manipulations on the turntables and the mixer, in order to drive peoples' attitude to the mix into many emotional states. Nowadays, when amateur-DJs can afford digital DJing systems to perform at home informally for their friends, the DJ music industry is focused on pushing these so called Bedroom-DJs [22] to the stage [9].

Mixing-DJs are essentially interested in the problem of beat-matching and cross-fading songs as *smoothly* as possible [25]. In that sense, we find that Mixing-DJs think about four main questions when they aim to introduce new music in their performances. These questions consider *What?*, *When?*, *Which?* and *How?* new music content will be introduced.

First, the performer needs to know *What?* songs will accompany well with the song being played. Second, knowing *When?* to introduce new music material lets the DJ manipulate the flow of the mix in order to drive the expectations of the audience. Third, the practice of beat-matching allows DJs to plan *Which?* song elements would sound more noticeable when playing different songs. Last, the DJs needs to know *How?* to seamlessly synchronize songs, a help that most currently available digital DJing systems[1] already provide.

In this paper we present *BeatJockey*, a tool for DJs that supports and enhances the traditional playback interaction of DJing. Besides from the traditional features already present in most current DJ systems, this system is also capable of suggesting and introducing music material, thus providing answers to the four aforementioned questions (*What?*, *When?*, *Which?* and *How?*). We believe that if the system is capable of playing back song slices preserving the event and rhythm structure of a master song then such a system will have acquired the basic knowledge of an experienced mixing-DJ, and therefore will be ready to help the non-experienced one. In order to do so, the system suggests beat-slices, taken from other songs of the playlist, that present similarities to the master song being played. These beat-slices form sequences that rhythmically match the master song. In addition, the system supports the synchronization of multiple songs, thus allowing coherent and rapid progressions between the songs in the mix.

BeatJockey uses beat-tracking [5] to help decide *When?* to playback beat-slices. A set of content descriptors [20] extracted from the songs, and machine learning techniques [11] indicate *What?* beat-slices could be played. In addition, *BeatJockey* uses a concatenative synthesis technique called Audio Mosaicing (AM) [17], in order to recreate a target sound by using slices from other sources. Our AM approach encodes *Which?* beat-slices should be sequentially played in order to resemble a master song. Lastly, the current *BeatJockey* prototype has been implemented in the Reactable [14] musical tangible user interface (TUI), changing consequently the normal way *How?* DJs manage to synchronize songs.

The remainder of this document is structured as follows: Section 2 overviews related systems that have contributed to DJing. Next, Section 3 presents some new possibilities for DJs to interact with songs, and describes the system's implementation and control. Finally, in Section 4 we evaluate how the users have assessed the music produced by our system.

[1]Stanton (Final Scratch), Rane (Scratch Live), Native Intruments (Traktor Scratch Pro)

2. RELATED WORK

This section introduces a set of systems under the scope of DJing. In the literature there are different approaches that solve individually the (*What?*, *When? Which?* and *How?*) problems of introducing music material. Accordingly, we classify these works with regard to the question they solve.

- *What?* songs would propperly accompany a given master song is addressed in [2, 15]. The authors present interfaces that take content descriptors into account in order to suggest new songs.
- *When?* to introduce sound events is addressed in [8, 12], in which the authors describe different synchronization strategies based on beat and tempo of the songs.
- The AM approaches by [17] and LoopMash[2] address *Which?* sequence of sound units should be sequentially played in order to resemble a given target sound. In addition to AM, different techniques for synthesizing a target sound out of pre-existing sound exist [21].
- In [9], the *How?* DJs are able to practice DJing with the help of devices is addressed. Some of these devices can replace the traditional vinyl[3], are intended to be used at home[4], or are oriented for gaming[5] [9]. Experimental DJing interfaces also exist that augment traditional equipments [1, 3, 19], while others aim to provide control over other performance parameters [18, 24]. In [6, 16, 23] some systems that offer novel ways of controlling the playback of songs are presented. In addition, a variety of systems supporting multi-touch interaction for DJs, either commercial[6] and research[7] oriented are found. In [10] support for DJing interaction is implemented under the Reactable[8], the same TUI used by*BeatJockey*.

We find however that none of these interfaces contemplates the *What?*, *When?*, *Which?* and *How?* problems at the same time.

3. A NEW PARADIGM OF SONG INTERACTION FOR THE DJ

The system we propose implements the basic set of standard DJ functionalities such as, playback of multiple songs, tempo adjustment, song's gain, filters, song positioning, etc.. Moreover, it also introduces new functionalities for enhancing DJs' creativity at their performances.

3.1 New functionalities

For every beat-slice of the master song the system tries to find a matching beat-slice from another song in the DJ's playlist. The suggested beat sounds will build beat sequences that resemble the master song. These beat sequences are sorted and disposed using AM. This results in a beat-mashed sequence that rhythmically accompanies the master song, and which the performer has the possibility to put in the foreground or leave in the background. Moreover, at any beat, the DJ has the possibility to drive the beat sequences synchronously towards any selected song.

With such functionalities, we speculate that the system might decrease the performance and cognitive load of experienced mixing-DJs, thus inciting and enhancing their creativity.

3.2 System's corpus

BeatJockey uses two layers of information extracted from the music in order to introduce a new paradigm of song interaction for DJing.

- The first layer of information provides a solution to know *When?* to trigger beat-slices. Since the *beat* is the event that the majority of people would follow in order to respond to the rhythm of the music [7], we have assumed that the *beat* is the basic cue that a DJ uses to synchronize songs. *BeatJockey* uses the algorithm BeatRoot developed by Dixon [5] to extract the beat times of the songs.
- The second layer of information solves *What?* sounds are more likely to sound coherent when played back together with a master song. In order to compare beat-slices of sound our system extracts two kinds of information from them. In [11], a collection of content descriptors useful to classify drum sounds is described. From this collection we have selected the following set: spectral energy, spectral spread and flatness, Mel-frequency cepstral coefficients and Bark-bands. These descriptors are computed with the help of an in-house library[9]. Second, we use a statistical model for labeling each of the beat-slices [11]. The model (support vector machine) classifies beat-slices into four different Percussive Class Labels (PCL), *Bass drum, Snare drum, Hi Hat* and *Cymbal* (BD, SD, HH, CY), that reflect which elements of a drum kit are more likely to be present in the beat.

The system's information corpus is illustrated in Figure 1. In this figure, the BeatRoot algorithm extracts the beat-times for each song of the DJ's playlist. Then, each beat is cut at the quarter-note level. For each beat-slice we then take the set of content descriptors previously mentioned, and their PCLs. The system packs the beat-slice information into their informational points. As these points' dimensions are formed by both high-level (PCLs) and low-level (content descriptors) information, they reflect information about the beat-slice sounds. This allows the system to find similarities between the beat sounds contained in the database, thus providing a solution for the questions of *What?* and *When?*.

3.3 System's performance modes

AM helps to suggest *Which?* sequences of beat-slices will be sequentially arranged and played, in order to maintain event-wise sychronization and *Percussive Structure* (PS), with respect to a master song. We define the PS of both units, a master song's fragment and the suggested beat-sequences, as the succession of their beat-slices's PCLs (e.g. (BD, HH); (HH,SD); (BD,CY)). Our AM approach always tries to preserve the PS's between the master song and the suggested beat-sequences (see figure 2).

1. In *Beat-mash* mode, the suggested beat-sequences are built from beat-slices of different selected songs. A

[2] `http://loopmash.com/`
[3] Technics (SL-DZ1200), Denon (DN-2500F), Vestax (Spin), M-Audio (Torq), Tonium (Pacemaker)
[4] Hercules DJ Control MP3
[5] Activision (DJ Hero), Arcade Games (Beatmania)
[6] Stanton (SCS.3D), `http://www.smithsonmartin.com/products/emulator/`, `http://www.algoriddim.com/djay-ipad`, `http://ipadmixr.com/`
[7] `http://www.soundwidgets.com/stribe/`
[8] `http://www.reactable.com`.
[9] `http://mtg.upf.edu/technologies/essentia`

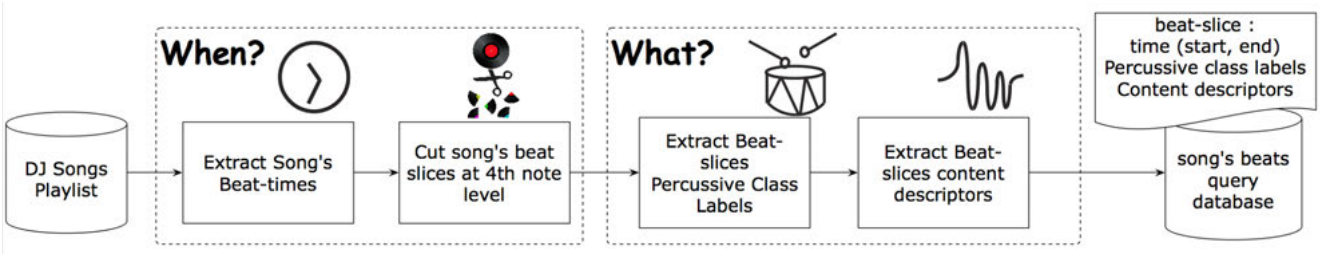


Figure 1: Two different infomation layers (*What?* and *When?*) are analyzed by the system to characterize music events in order to suggest music material.

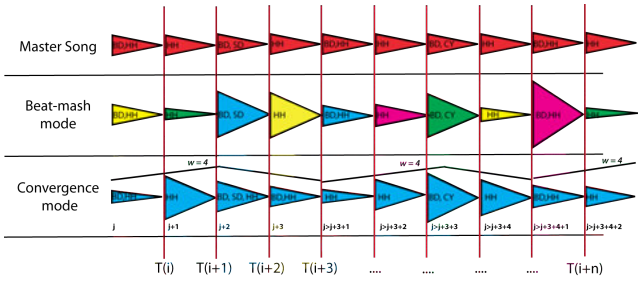


Figure 2: An overview of the new song interaction functionalities. Vertical lines depict the master-song's beat-times, where (i) denotes the beat number of a song with the unit being the quarter-note. Color denotes the different songs in the DJ's playlist. The area of the triangle denotes the similarity of the suggested beat-slices with respect to the master-song's beats. The first row illustrates a master song decomposed in beat-slices of sound that are played sequentially. The second row corresponds to the *Beat-mash* mode and the third row to the *Convergence* mode. Here j denotes the target song's beats indexes and w a PS query-window's size. See text for further details.

Nearest Neighbor (NN) algorithm [4], with euclidean distance is used to search for similar beat-slices. Given a target beat-slice, its most similar slice will be the one containing strictly the same PCLs and being the closest, in euclidean distance, according to its content descriptors.

2. In *Convergence* mode the intention is to converge the rhythms between the master and one selected target song. In that case, beat-sequences of w beats with the same PS of the master song, will be searched. The beat-sequences are compounds of beat-slices with increasing beat indexes in the target song. This assures a progression towards the target song, while keeping an automatic synchronization between the master and a target song. The user may change the target song at any beat time. Consequently, this mode allows for rapid and coherent progressions towards different songs of a playlist.

In both modes, when the system does not find a similar beat-slice or a target beat-sequence, it does not suggest any suggestion at this particular moment.

3.4 System's control

In order to reinforce the How? problem, we have implemented *BeatJockey* within the Reactable application [13], as we believe that both sides can win from this symbiosis. *BeatJockey* has been designed taking into account both the specific affordances of this device as well as the prevalent turntable metaphor, nevertheless its main functioning principles could be easily ported to other interfaces. On one hand, *BeatJockey* extends the limited DJ interaction that Reactable implements. On the other hand, the Reactable's multitouch tangible interface provides affordances comparable to analog-digital devices. It allows different users to perform in the same interface by sharing the controls on the surface, and it also offers a modular approach that eases the inclusion of new features, as long as they adhere to the interface's main metaphors.

Reactable offers four different types of objects with varied typologies: sound generation objects; sound processing objects; control objects; and global objects, that modify parameters affecting all the objects in the table. The functioning principle of these objects are the same for all four types. Objects are activated when they are put on the table's surface, and the object's control parameters may be modified by rotating them, and by moving virtual sliders or selectors around the objects, with the fingers.

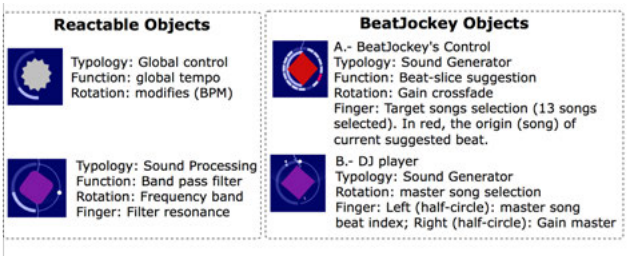


Figure 3: Overview of *BeatJokey's* control objects.

With these considerations, we have designed the control objects shown in Figure 3. Their sliders and the selectors provide affordances comparable to the classical DJ setup, although the positioning of a track is centered at the beat time. The system does not support yet time-stretching of songs and beat-sequences.

The performance modes[10], *Convergence* (one song), and *Beat-mash* (more than one song), are activated by selecting songs with the controller A. If no target songs are selected the system does not suggest any beat. If object A is taken out of the table and put back, the selection of target songs will be reset (no songs selected). There may be different users using combinations of A and B objects, thus allowing for multi-DJ collaborative performances.

4. EVALUATION

We have not yet evaluated *BeatJockey* as a live tool, but as a proof of concept we have done a preliminary evalua-

[10]please refer to video, http://dl.dropbox.com/u/13952105/BeatJockey.mov

tion of *BeatJockey*'s performance, asking ten listeners to listen to and evaluate the results of previously recorded *BeatJockey* sessions. Evaluation results for the *Beat-mash* mode reflected that the suggested beat-slices were preferred over randomly generated ones (t-test p-value<0.05). For the *Convergence* mode, evaluation results were not statistically significant (t-test p-value = 0.782) for determining the most appropriate window size (w = 2, 4 and 6 were tested). Nevertheless, we find that when w is too small (w<=2), the target song does not progress and stays in a beat-sequence that matches perfectly the master's PS. Conversely, when too large values of w are used (w>=6), the target beat-sequences are not found, and the system does not preserve the continuity of the suggested beat-sequence. Therefore, our final implementation uses a $w = 4$.

5. CONCLUSIONS

We have overviewed current trends in the development of DJ supporting systems, and we have introduced *BeatJockey*, a system which takes into account the basic DJs' playing rules. With these rules, *BeatJockey* is able to support non-experienced mixing-DJs while it also provides to more experienced DJs, new ways to interact with songs. *BeatJockey* also extends the Reactable functionalities taking benefits from the Reactable's main functioning principles.

BeatJockey needs further refinement. The mapping between Reactable objects and system functions needs to be further studied and improved in order to achieve a better 'turntable' metaphor. Moreover, in order to avoid silent beat-slice suggestions, we need to allow the system to provide more flexible matchings (e.g. not taking into account PCLs) and also let the user control the w parameter.

The interface is yet to be evaluated with both expert DJs and novice users. This will be done in the near future. An online implementation of the analysis stage could help to synchronize DJ sessions between different performances at different places. We think, this online implementation would provide a rich space of interaction between multiple performers and audiences.

6. REFERENCES

[1] T. Andersen. Mixxx: Towards novel DJ interfaces. In *Proc. of NIME*, pages 30–35. National University of Singapore Singapore, Singapore, 2003.

[2] D. Baur, T. Langer, and A. Butz. Shades of music: Letting users discover sub-song similarities. *ISMIR*, 2009.

[3] T. Beamish, K. Van Den Doel, K. MacLean, and S. Fels. D'groove: A haptic turntable for digital audio control. *Proc. of ICAD, Boston, MA*, 2003.

[4] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? *Database Theory—ICDT'99*, pages 217–235, 1999.

[5] S. Dixon. Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research*, 36(1):39–50, 2007.

[6] K. Fukuchi. Multi-Track Scratch Player on a Multi-Touch Sensing Device. *Entertainment Computing–ICEC 2007*, pages 211–218, 2007.

[7] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–54, 2005.

[8] G. Griffin, Y. Kim, D. Turnbull, and P. Swarthmore. Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups. *ICASSP*, 2010.

[9] K. Hansen. The acoustics and performance of DJ scratching. *Doctoral Thesis Stockholm, Sweden*, 2010.

[10] K. Hansen and M. Alonso. More DJ techniques on the reactable. In *Proc. of NIME*, 2008.

[11] M. Haro and P. Herrera. From low-level to song-level percussion descriptors of polyphonic music. In *International Conference on Music Information Retrieval, Kobe, Japan*, 2009.

[12] H. Ishizaki, K. Hoashi, and Y. Takishima. Full-automatic DJ mixing system with optimal tempo adjustment based on measurement function of user discomfort. *ISMIR*, 2009.

[13] S. Jorda. On stage: the reactable and other musical tangibles go real. *International Journal of Arts and Technology*, 1(3):268–287, 2008.

[14] S. Jorda, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reactable*. In *Proc. of ICMC, Barcelona, Spain*, pages 579–582, 2005.

[15] A. Kapur, R. McWalter, and G. Tzanetakis. New Music Interfaces for Rhythm-Based Retrieval. In *Proceedings of the 6th International Conference on Music Information Retrieval*. Citeseer, 2005.

[16] S. Kiser. spinCycle: a color-tracking turntable sequencer. In *Proc. of NIME*, pages 75–76. IRCAM—Centre Pompidou, 2006.

[17] A. Lazier and P. Cook. MOSIEVIUS: Feature driven interactive audio mosaicing. In *Digital Audio Effects (DAFx)*, 2003.

[18] T. Lippit. Turntable music in the digital era: designing alternative tools for new turntable expression. In *Proc. of NIME*, pages 71–74. IRCAM—Centre Pompidou, 2006.

[19] A. Pabst and R. Walk. Augmenting a rugged standard DJ turntable with a tangible interface for music browsing and playback manipulation. In *3rd International Conference on Intelligent Environments, 2007. IE 07.*, pages 533–535. IET, 2008.

[20] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO IST Project Report*, pages 1–25, 2004.

[21] D. Schwarz, G. Beller, B. Verbrugghe, S. Britton, et al. Real-time corpus-based concatenative synthesis with catart. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06),(Montreal, Quebec, Canada)*, pages 279–282. Citeseer, 2006.

[22] P. Terrett. *Bedroom DJ: a beginner's guide*. Omnibus Pr & Schirmer Trade Books, 2003.

[23] Y. Tomibayashi, Y. Takegawa, T. Terada, and M. Tsukamoto. Wearable DJ system: a new motion-controlled DJ system. In *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, pages 132–139. ACM, 2009.

[24] N. Villar, H. Gellersen, M. Jervis, and A. Lang. The ColorDex DJ system: a new interface for live music mixing. In *Proc. of NIME*, page 269. ACM, 2007.

[25] S. Webber. *DJ Skills: The essential guide to Mixing and Scratching*. Focal Press, 2007.

Traces – Body, Motion and Sound

Jan C. Schacher
Zurich University of the Arts
Institute for Computer Music and Sound Technology
Baslerstrasse 30 8048 Zurich, Switzerland
jan.schacher@zhdk.ch

Angela Stoecklin
The Fusion Projects
Neugasse 33
8005 Zürich, Switzerland
an.stoecklin@bluewin.ch

ABSTRACT
In this paper the relationship between body, motion and sound is addressed. The comparison with traditional instruments and dance is shown with regards to basic types of motion. The difference between gesture and movement is outlined and some of the models used in dance for structuring motion sequences are described. In order to identify expressive aspects of motion sequences a test scenario is devised. After the description of the methods and tools used in a series of measurements, two types of data-display are shown and the applied in the interpretation. One salient feature is recognized and put into perspective with regards to movement and gestalt perception. Finally the merits of the technical means that were applied are compared and a model-based approach to motion-sound mapping is proposed.

Keywords
Interactive Dance, Motion and Gesture, Sonification, Motion Perception, Mapping

1. INTRODUCTION
In this publication the question of motion analysis and mapping is regarded from a very specific angle. Starting from the experience of a contemporary improvising dancer, the issues of motion, gesture, flow and force are addressed. When dealing with elements that characterize a dance movement terms such as the motion description fundamentals start to appear: inertia, energy, space and temporal structure but also terms of expressive potential and of anticipation, perception and recognition of specific motion patterns. In an attempt to better understand these fundamentals a scenario for interactive dance that originates from a real-life artistic process is identified and defines a small exploratory study. A number of measurement techniques are brought to bear on a constrained set of movements, with a specific question in mind. The movements and the measured data are combined in an audification and sonification process, as well as in different technical visualisations. On a first level the differences between measurement techniques become apparent, since the underlying physical phenomena are directly informing the results. On a second, higher level of complexity and correlation it is less the direct relationships between the measured streams of data that are interesting, but – via the translation into a different sensory mode – the emerging salient features of movement or even gesture in dance.

2. BACKGROUND
Relating motion, actions and gestures to musical processes is an essential part of developing interaction systems involving technologically mediated instruments. One central aspect in this relationship is the mapping strategies applied. [14] A further important aspect is the type of interfacing / gesture acquisition technologies utilized, since this influences the type of information obtained. The NIME community's main focus lies in this area. A number of attempts have been made to classify these devices [11] and create a comprehensive overview of the affordances they offer. [9] Many of the projects presented in this context explore the capabilities of the usually most advanced technical solutions available. These works generate know-how about the application of these devices, their strengths and weaknesses for musical use – usually such a device is designed for a different context – and characterizes precisely the type of information generated.
The traditional music performance with instruments builds upon a relationship with sound directly through a physically sounding object (except for the voice, where the body is sounding directly). The schooling of instrumentalists involves a great deal of body conditioning and training, or imprinting of fine motor skills [6] related to and occurring in an adaptive loop with the production of sound. The movements and actions used for this task are almost completely informed by the physics of the instrument. Economy of motion is a guiding principle only to be transgressed when internal impulses demand expression. [15] In general four types of movement can be distinguished: reflex, locomotive, instrumental, and expressive movement. Musical actions are therefore essentially instrumental, and only a small percentage actually becomes expressive.
The term musical gesture is sometimes used in this context, without actually making a clear distinction between instrumental and expressive motion. In other fields, such as communications theory and linguistics, gestures denote a very specific type of motion. It is considered "an expressive movement that is not consciously thought out beforehand" [4] and serves to enhance thought and communication. Gestures also carry a signification: "Gestures are not just movements and can never be fully explained in purely kinetic terms. They are not just arms waving in the air but *symbols* that exhibit meaning in their own right". [10] This quality, which is present in the expressive part of music-related movement, has to be differentiated clearly. A term that originates from linguistics and which highlights a difficult challenge for motion analysis and mapping is that of co-articulation. [7]
In contemporary dance, however, gestures are considered higher-level expressive entities that convey more than just movement. It is important to understand the differing views between dance and music performance in this regard. Unlike in film, contemporary dance attempts to render movement into something abstract and detached from everyday connotations and situations. These abstract dance-movements represent

traces of physical but also mental processes concerning the body in space. A trained dancer learns to circumvent everyday movements, to detach herself from them and to create new patterns and variations thereof. The motivation for movement might be musical or visual, although the intention does not always include the projected image of the body. Musical elements such as rhythm and pulse play a central role in structuring motion in dance. Most modern dance notations, after Laban [12], are foremost descriptive and not expressive. The main categories described in these dance-languages are energy, placements and motion paths of body parts, placement in space and shape of body motion with regards to physical properties such as momentum and inertia. An arm movement for example might be described as a circular movement going upwards to the tipping point, then swinging with its own momentum down and back in a pendulum arch, then swinging to the front with the remaining inertial energy.

Dance gestures on the one hand are always tied to their body, which is at the same time their medium and shows their result and are only quite recently being measured, stored and analysed with technical means. Music on the other hand can very well exist without a body, especially in technically stored forms. Furthermore musician execute their instrumental movements adding some expressive parts without ever consciously balancing the two, thus the level of abstraction lies in the music an not the motion. The question arises now about how to identify expressive elements of motion between dance and music performance, where to look and what categories to apply.

3. SCENARIO

An interesting case arises, when a dancer is put into an interactive situation, taking on the role of a musician, so to speak. The motivation for movement might stay the same but the rhythmic and dynamic execution changes, when sound is produced or controlled by movements. The scenario devised starts from the idea that a dancer will perform a dance sequence consisting of a chain of gestures that can be chunked into movement elements. In order to gain more precise information the situation is a reduction to a few core aspects and consists of short twenty-second phrases covering a limited space horizontally as well as vertically. The dancer choreographs the sequence and executes it numerous times while varying characteristics such as intensity, speed and effort. Unlike a more classic live-electronic approach [3], here the music is generated *after* the fact, there is no sound during the performance, the dancer is only following her inner rhythm not some exterior material. In order to avoid an excess of data and to be able to compare the different sensors used, the motion-capture is constrained to two marker-groups, one on each wrist, mirroring the accelerometers placed on the body.

Figure 1. The dancer in our lab wearing accelerator bracelets and motion-capture markers on her wrists. The insert shows one of the bracelets in combination with a rigid-body marker used for motion capture.

4. METHOD

The measurement technologies we use range from simple accelerometer bracelets, to more complex inertial measurement units, from frontal two-dimensional video tracking with classic image analysis to an eight-camera marker-based motion-capture system. Each of these techniques offers a specific perspective on the dancer's body. We chose to use them simultaneously because they represent on the one hand a rich palette of tools, and on the other hand we hope that the measurements can tell us something about the accuracy and performance of each system and might permit a qualitative comparison between the different measuring techniques. In the following section, the different sensors and their stage-worthiness are briefly described.

4.1 Sensors

The wireless sensor bracelets were described in detail in an earlier publication. [14] They consist of a three-dimensional accelerometer and also provide two dimensions of gyroscopes. The update rate is between 50 and 100 Hz. The wireless inertial measurement unit (IMU) provides three orthogonal data streams for each measurement type: acceleration, gyroscope rotation and magnetic orientation. These values can be combined to obtain an absolute reference heading value and the absolute attitude of the sensor. This information is interesting mainly with regards to the overall body attitude. The update rate is between 50 and 100 Hz. These two sensor are stage worthy and applicable to a variety of scenarios.

The frontal two-dimensional video camera is used for body silhouette and lateral spatial analysis. By using an industrial firewire camera sufficiently high frame rates are obtained to be useful in comparison with the other sensors. The update rate can go from 60 to almost 100 Hz. Since it uses traditional background subtraction techniques this system is very light dependant. It is only useful in stage situations where absolute control over the lighting can be exerted (see Figure 2.). [2]

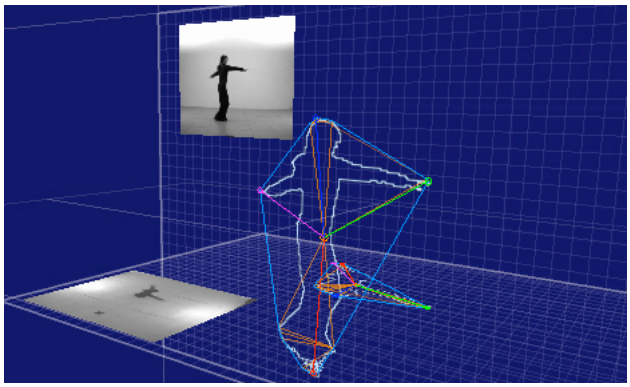


Figure 2. Dual camera silhouette analysis.

The motion-capture system we use is a smaller commercial eight-camera system. The three-marker solid bodies work very well in a space of the size of our lab (see Figure 1.). Although the system is one of the new generations of game-derived systems it delivers 100 frames per second and sub-millimetre precision. Since this is the biggest of the sensor-systems its application in a stage context poses a bigger challenge. [16]

4.2 Software

A variety of software applications are used to gather, store and analyse the sensor streams and their data. Apart from the commercial motion-capture system all the acquisition software was developed specifically for these sensors. The wireless sensors and their serial streams are parsed and transformed to OSC with dedicated proxy servers written in C++ using

openFrameworks. [17] The frontal camera input and analysis is using Jitter and some custom code to calculate the convex hull and cardinal points of the silhouettes. [14] The motion-capture system runs its own software package on a dedicated windows machine and needs it's a proxy to translate from its native Nat NET protocol to OSC. This is a dedicated command line application written in C++. The data time tagging and storage is implemented with the Jamoma [13] module for GDIF [8] recording, which is based on FTM [1] and writes SDIF files. These modules run within MaxMSP, where all subsequent audification (straight data-to-sound mapping) , sonification (re-interpretation of data into sound), visualisations and calculations are executed.

4.3 Data Analysis

The resulting streams of values contain single integers and up to ten floating-point values. A central clock synchronizes the entire recording, and all data is recorded for each frame. In addition a video is recorded with the frame numbers inset for future synchronisation at playback. In order to reduce complexity further and to focus the analysis on a clearly perceivable element, only the accelerations from the motion-capture system and the wearable sensor bracelets are used.

The position data from the motion-capture system is analysed to its first two derivatives. These are purely spatial traces or kinematic calculations, no forces or masses are taken into account.

The accelerometer measurements are transformed to their summed absolute values, which only represents energy. In this form the values contain no more spatial or directional information. This data type differs from the former as it represents masses – actually a real micro-mass within the sensor – and the forces exerted onto them. Finally, in order to be able compare the two types of acceleration values we normalise them.

The visualisation tries to reintegrate as much information as possible into images and graphs. In an attempt to fuse spatial and temporal dimensions of the collected data, two graphical solutions – the timeline and the 3d representation with trails – are chosen. The actual video and imagery of the silhouette is added, since they enhance the perception, especially when viewed as moving images. The illustrations in this publication try to convey a sense of the temporally evolving values.

The audification uses filtered noise. The band pass filter is controlled on the frequency domain by the horizontal spatial x-axis, the output gain by the horizontal spatial y-axis and the steepness by the sum of acceleration on the bracelet. This algorithm presents a very basic one-to-one mapping but gives a clear sonic rendering of the motion and acceleration trajectories. (see Figure 5. upper half)

The sonification uses a granular cloud, where all spatial parameters obtained from motion capture are applied to the grain parameters and all accelerometer values influence the sample playback and windowing of the grains. (See Figure 5. lower half)

5. RESULTS

After combining the relevant streams and their cooked or analysed form into a range that puts them on the same levels, the comparative evaluation begins. Since the main question addresses qualitative rather than quantitative measures, no absolute numerical comparison was undertaken. The following illustrations show two frames from the motion sequence. This version of the sequence was executed in moderate dynamics with normal speed, so that the measured values vary moderately within their given ranges. The main feature visible is the circular motions of both wrists in space. The spikes in the accelerations corresponds to the changes in direction, which is clearly visible in the first frame's yellow line (Figure 3. upper half) and in the very last peak on the first time plot and the spike located at around 17.9 seconds immediately before the highest point in the second plot. (See Figure 3. lower half)

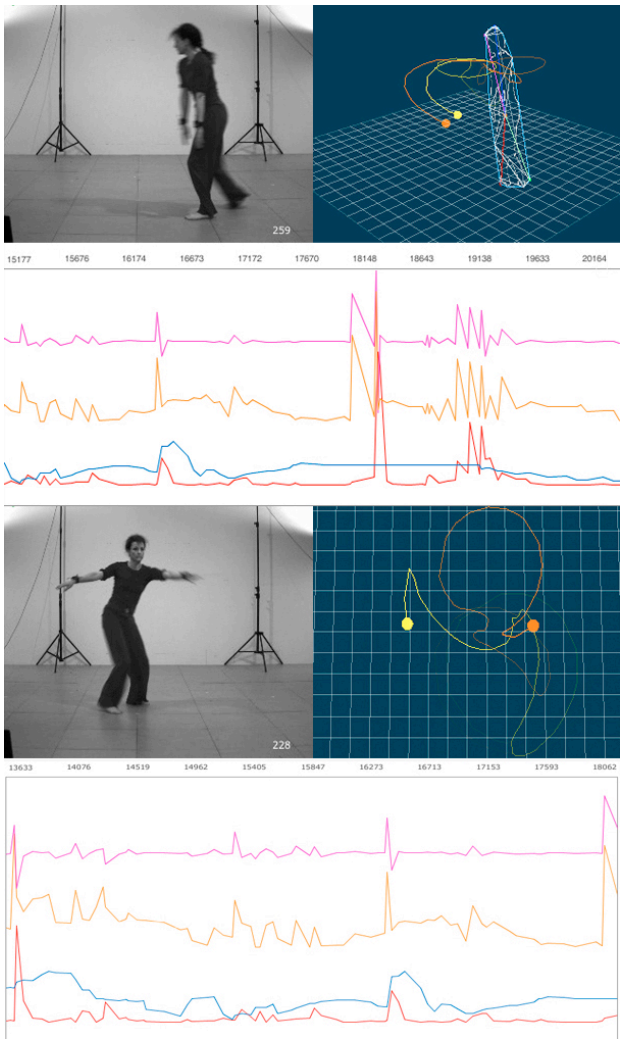


Figure 3. Visualisation of motion traces and wrist accelerations. The red trace shows the absolute kinematic acceleration, the blue trace depicts the dynamic acceleration.

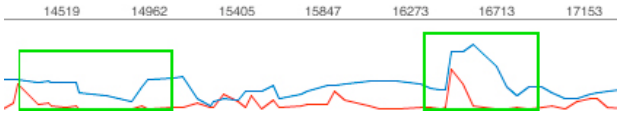


Figure 4. The red line depicts kinematic acceleration extracted from the motion capture; the blue line shows the accelerometer values. Note the overhang in the green boxes.

Considering the difference between the two measurement technologies, one being purely kinematic, the other a dynamic mass, there are obvious differences on how the values evolve. Where the kinetic movement stops, the dynamic mass in the accelerometer still has momentum to dissipate and goes into a negative acceleration with its direction vector inversed. As can be seen around 14.5 seconds in (Figure 4.) the inertial decrease of movement continues into the next onset and is carried over an even small absolute movement. At 16.3 seconds a very clear coupling of a jerky movement with a delayed but parallel decay

in the dynamic mass is visible. Here again, the next small displacement is answered by a compensatory acceleration of the inertial sensor with the inverted direction vector.
The auralisation strategy demonstrates that low-level linking of parameters from the motion to the sound domains works well. The measured energy of the movement, particularly the inertial mass of the accelerometer translates well into sound energy as applied to this simple subtractive synthesis. (Figure 5. upper half)

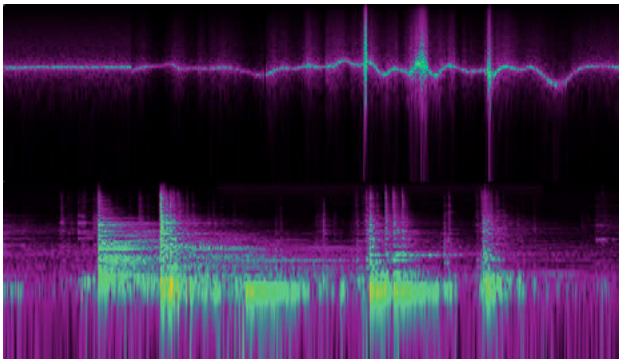


Figure 5. Sonogram of the audification of the motion fragment and of the sonification via granular synthesis.

This unsurprising fact is put into a different perspective, when a more complex sound generation algorithms are applied. Highly non-linear algorithms such as FM-synthesis or highly parallel processes such as granular synthesis offer a richer and more diverse sonic result. The granular synthesis algorithm used here produces a much richer sonic experience, but is less easily measured in terms of spectral content. (Figure 5. lower half)
Both hearing-based methods of data-display afford a perception of the motion sequences that emphasises the gestalt. The difference between the two methods, from the purely parametric mapping to a more subjective interpretation changes the richness of the sonic output.

6. DISCUSSION AND CONCLUSION

Comparing sensor data from purely kinematic measurement and an inertial mass sensor is obviously a strong reduction of information concerning a dancer's movement. Isolating and equalising two streams of values coming from the wrist of the dancer confirm a salient feature. The difference between kinematic and dynamic inertial measurement – but also the application of audification and sonification – shows that the physical properties of the moving body, such as its mass, momentum and inertia are more likely to be perceived as the carriers of expression. The energy or effort expended in the movement, which is the main category the dancer uses for creating and structuring a choreography, becomes more clearly visible in the data of the inertial sensor than the absolute spatial position acquired through the motion-capture system. The elements that comprise a gesture rather than a movement seem not to be clearly accessible in the data-streams, even though segmentation or chunking [5] is easily achieved at the rest-points of spatial, kinematic motion. This would indicate that in order to more naturally reflect the dynamic states of the body – which is what our perception principally anticipates and interprets – a physical model of the body should be introduced as a mapping category. This exploratory investigation also indicates that on-body sensors with their egocentric perspective remain a valid tool even when compared to the allocentric visual motion acquisition systems. When focusing on the perception of motion through other channels than the purely visual mode, the kinematic traces in space do not represent our motion as well is implied by the technology. The combination of the different types of sensor information with a model-based approach seems to promise the richest translation possibilities for body motion to sound.

7. REFERENCES

[1] Bevilacqua, F. Müller, R., Schnell, N. MnM: a Max/MSP mapping toolbox. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression*, Vancouver, BC, Canada.

[2] Camurri, A., S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, G. Volpe. Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. In *Computer Music Journal.,* 24(1):57–69, 2000.

[3] Eckel, G., Pirro`, D., Sharma, G, K. Motion-enabled live Electronics. In *Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference*, 23-25 July 2009, Porto.

[4] Gallagher, S. (2005) *How the Body Shapes the Mind,* Clarendon Press, Oxford.

[5] Godøy, R. I. *Systematic and Comparative Musicology: Concepts, Methods, Findings*, Chapter Reflections on Chunking in Music, pp. 117–132. Peter Lang, 2008.

[6] Godøy, R.I., Motor-Mimetic Music Cognition. In *Leonardo, Vol. 36, No. 4 (2003)*, pp. 317-319, MIT Press.

[7] Godøy, R.I., Gestural Imagery in the Service of Musical Imagery. In A. Camurri and G. Volpe (Eds.): *GW 2003, LNAI 2915*, 2004. Springer-Verlag Berlin Heidelberg.

[8] Jensenius, A.R. (2007) *Action – Sound, Developing Methods and Tools to Study Music-Related Body Movement*. Ph.D. Thesis, Department of Musicology University of Oslo.

[9] Magnusson, T. An Epistemic Dimension Space for Musical Devices. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, Sydney, Australia.

[10] McNeill, D. (1992) *Hand and Mind: What Gestures Reveal about Thought* Chicago, University of Chicago Press.

[11] Miranda, E.R, Wanderley, M.M. (2006) *New digital musical instruments: control and interaction beyond the keyboard.* A-R Editions, Inc.

[12] Laban, R. and F.C.Lawrence, (1974) *Effort: Economy of human movement,* second edition, MacDonald & Evans Ltd.

[13] Place, T. Lossius, T. Jamoma: A Modular Standard For Structuring Patches In Max. In *Proceedings of the International Conference on Computer Music* (ICMC'06) New Orleans, USA, 2006.

[14] Schacher, J.C. Motion To Gesture To Sound: Mapping For Interactive Dance. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, Sydney, Australia.

[15] Wanderley, M. M., B. W. Vines, N. Middleton, C. McKay, and W. Hatch. The Musical significance of clarinettists'' ancillary gestures: An exploration of the field. *Journal of New Music Research*, 34(1):97–113, 2005.

[16] http://www.naturalpoint.com/optitrack/

[17] http://www.openframeworks.cc/

URLs accessed in May 2011

MoodMixer: EEG-based Collaborative Sonification

Grace Leslie
Department of Music and
Swartz Center for Computational Neuroscience
University of California, San Diego
gleslie@ucsd.edu

Tim Mullen
Department of Cognitive Science and
Swartz Center for Computational Neuroscience
University of California, San Diego
tmullen@ucsd.edu

ABSTRACT

MoodMixer is an interactive installation in which participants collaboratively navigate a two-dimensional music space by manipulating their cognitive state and conveying this state via wearable Electroencephalography (EEG) technology. The participants can choose to actively manipulate or passively convey their cognitive state depending on their desired approach and experience level. A four-channel electronic music mixture continuously conveys the participants' expressed cognitive states while a colored visualization of their locations on a two-dimensional projection of cognitive state attributes aids their navigation through the space. *MoodMixer* is a collaborative experience that incorporates aspects of both passive and active EEG sonification and performance art. We discuss the technical design of the installation and place its collaborative sonification aesthetic design within the context of existing EEG-based music and art.

Keywords

EEG, BCMI, collaboration, sonification, visualization

1. INTRODUCTION

Alvin Lucier's *Music for Solo Performer* was premiered in 1965 at Brandeis University. Widely considered the first live brainwave music performance, it represented a break from the electronic music performance tradition of the time, and remains unique for a number of reasons. First, the performer's alpha (8-12.5 Hz) brainwaves drive percussion instruments directly through coupled amplifiers, allowing him to generate real acoustic events (not synthesized ones) at roughly 10 Hz. The resulting "acoustic" sound material contrasts with the synthesized or spliced concrète styles of electronic music making of the time. Even today, Solo Performer remains somewhat distinct as an electronic performance in its physically manifest yet acousmatic materials.

A second, more subtle and important distinction that *Solo Performer* holds, at least in 35 years of hindsight, lies in the active and purposeful nature by which the performer modulates his attentional state to produce these sound events. Approaches that we would now describe as sonification were common in the tape music community of the time, but Lucier avoided these in favor of a more active approach. Lucier describes his response in a 1981 interview,

> "... most of my colleagues at Brandeis said, "Oh, that's a wonderful idea. You ought to tape record it, speed the sounds of the brain waves up, slow them down, reverberate them, filter them".... I had to eliminate those [techniques] in order to get at the poetry of the piece, which demanded that a solo performer sit in front of an audience and try to get in that alpha state and to make his or her brain waves come out.[6]"

Thirty-eight years later, Steve Mann, James Fung, Ariel Garten and Chris Aimone produced the *Regen/DECONcert* series in which 48 participants donned wearable EEG hardware to manipulate musical parameters of a jazz ensemble performance [8]. The performers were not given any explicit instructions as to how they should manipulate their cognitive state; rather, the collective alpha activity of the population was mapped onto musical parameters directly.

These two pieces sit on the extremes of a passive-active sonification axis; while *Solo Performer* represents an active approach in which the performer is consciously manipulating his or her brain state to reach a desired musical effect, *Regen* embodies a passive approach where participants' EEG is used for musical purposes irrespective of any direct, conscious control on their part.

The passive-active continuum in EEG sonification systems – often referred to as brain-computer music interfaces (BCMI) [9] – can be seen as a specific instance of a more general passive-active-reactive categorization scheme recently proposed within the brain computer interface (BCI) community [13]. A *passive* BCI is one in which the cognitive state of the individual is unobtrusively "monitored" without conscious control on the part of the individual. Feedback is not a necessary component. An *active* BCI is a closed-loop system which derives its outputs from brain activity which is directly and voluntarily controlled by the user, independently from external events, with the intention of controlling an application. Real-time feedback indicating the current output state of the system is generally an essential component. A *reactive* BCI is similar to an active BCI. Here the system measures the neural response to external stimulation. The user exerts control over the system by voluntarily directing his attention or otherwise indirectly controling how the brain processes the external stimuli. One example of a reactive BCI system as a music interface is Mick Grierson's adaptation of a standard "P300 Speller" BCI to allow an individual to compose musical note sequences by selectively attending to different symbols randomly illuminated on a computer display. When the attended symbol is briefly illuminated, the brain generates a neural response, which is detected in the EEG, and used to produce an associated note [3].

Solo Performer and *Regen* also exemplify two extremes of a solo-collaborative sonification axis. *Solo Performer*, like many contemporary BCMI systems, involves a single individual interacting with the system in isolation. EEG-based musical performances with multiple performers, while far less common than solo performances, date back to Rosenboom's 1971 *Ecology of The Skin* [12]. Here EEG alpha power from ten observer-participants was used to control ten parts of a synthesized sound texture. This is an early example of a collaborative BCMI in which the state of the system is determined by the neural state of two or more individuals. More recently, Miranda and Brouse proposed a "collaborative audification" approach in their Internet-enabled *InterHarmonium* project which introduces additional possibilities for large-scale collaborative BCMIs [9]. *Regen* embodied an extreme end of the collaborative sonification spectrum wherein the state of the music interface was determined by the collective neural activity of a large number of people.

Our *MoodMixer* project was conceived to incorporate aspects of both the active and passive approaches to EEG-based music creation while using hardware, data treatments, and electroacoustic and visualization techniques to facilitate a multiuser, collaborative approach.

2. COLLABORATIVE SONIFICATION

The performance instructions for the installation are as follows: The two participants sit in a room, each wearing a comfortable wireless EEG headset, as depicted in Figure 1. From each EEG data stream, two indices, each measuring a different aspect of the participant's cognitive state, are calculated. By mapping each measured state onto a one-dimensional axis, each participant is able to independently "navigate" within a shared two-dimensional musical interface; in this way, the musical aesthetic at a given point in time is determined by the combined cognitive state of both participants. In this instantiation, location along the ordinate (x-axis) is determined by a calculated index that roughly corresponds to the degree of relaxation or "meditation", while location along the abscissa (y-axis) is determined by an index corresponding to the participant's level of sustained attention or "focus." Another active control option is provided, as sound samples and visual flashes may be triggered by individual eye blinks or predefined blink sequences.

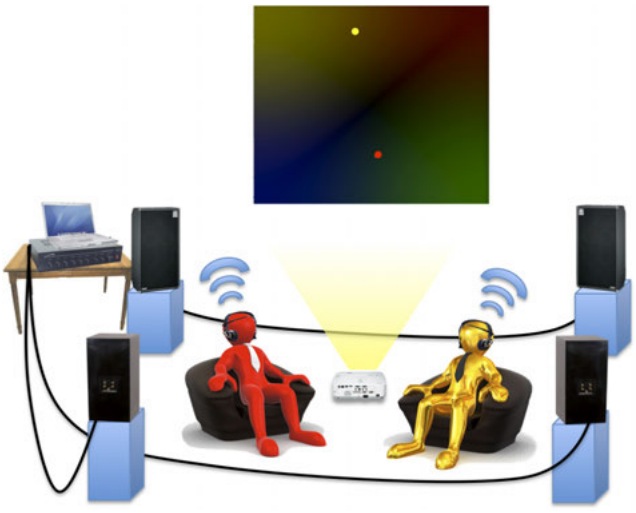


Figure 1: Depiction of the installation in use.

Novice participants may wish to adopt an observational relationship with the system as their changing cognitive state is passively monitored and represented in the audio-visual landscape. As participants use the feedback to gain experience in manipulating their neural state and actively controlling the interface they may shift to an "active" regime where they may choose to "improvise" and independently experiment with combinations of music drawn from two different regions of the music landscape. Alternately, participants may choose to attempt to cooperate closely with each other in creating a composition by mirroring the other's affective state (e.g., if one participant is in a relaxed, but focused state, the other should attempt to do likewise). Participants may define "games" which they play with each other. For instance, during a "calm" interval one participant might execute a sequence of blinks triggering a sound sample or visual pulse intended to induce dramatic changes in the arousal of the other participant, and thereby a dramatic shift in the evolving composition. Real-time visual feedback of each participant's current position (represented as a colored cursor) in the two-dimensional musical landscape affords an increased sense of control and interactive engagement in the compositional process.

2.1 Technical Design

The technical architecture of the system is depicted in Figure 2. In our first implementations we have used the Neurosky MindSet™, a relatively low-cost (< $200) wearable EEG system[1]. As seen in Figure 2, the system features a "headset" design with a single active electrode (left or right prefrontal cortex), as well as reference and ground electrodes on the earlobes. The headsets utilize "dry" (gel-free) sensing technology and feature integrated Bluetooth, alowing data to be streamed wirelessly to a laptop. Raw EEG data from the single electrode is sampled at a rate of 512 Hz. Spectral power in the delta (0.1-3 Hz), theta (4-7), alpha (8-12 Hz), low, midrange, and high beta (12-15, 16-20, and 21-30 Hz, respectively) are calculated on the headset once per second using fast fourier transforms. Two cognitive state indices ("focus" and "meditation/relaxation") are then calculated using specific combinations of these bandpower features intended to correlate with the degree of focused attention/cognitive load and meditation/relaxation [1, 11]. Although the specific combination of features is proprietary to Neurosky, it is well-known that frontal alpha power is positively correlated with relaxation and/or meditative states of mind, while frontal beta power is positively correlated with increased concentration and focus. Experimenting with our own frequency ratios confirmed this. However, Neurosky's indices are based on a large normative dataset and thus provide immediately useable (if only approximate) normalized (0-100%) estimates of the indicated cognitive state without the need to collect any calibration data. The raw EEG data stream and relaxation and focus indices are streamed into a Max/MSP/Jitter patch via a Bluetooth connection using an external by Kyle Machulis[2]. The index values are then smoothed using a four-second moving average. The smoothed indices are used to control a four-way equal loudness panner which assigns the relative loudness "weights" of four audio tracks, each containing music material representing a combination of two of the cognitive index extremes (e.g., high focus, low relaxation). The music mix is projected via a four-channel surround sound system to create a spatial representation of the participant's mental state.

MoodMixer incorporates a two-dimensional colored visual

[1] http://www.neurosky.com
[2] http://www.nonpolynomial.com

representation (implemented in Jitter) in which each vertex of a square represents a combination of extreme values of the two cognitive indices. The square is comprised of a weighted mixture of four colored gradients, each associated with one vertex of the square, the weights of which are determined by the four-way panning curve. Each participant is given control of a uniquely-colored cursor, indicating his or her position in the two-dimensional cognitive landscape. Specifically, the participant's two cognitive indices are respectively mapped to the x- and y- coordinates of the cursor. Thus, as each cursor approaches a given vertex, the luminosity of the associated color gradient is smoothly increased (and that of other vertices proportionately decreased) in accordance with the levels of the associated cognitive indices. This approach is conceptually similar to Jacqueline Humbert's 1974 *Brainwave Etch-A-Sketch* in which two individuals, by manipulating their alpha power, each control the x or y position of a point of light on two-dimensional analog interactive display [12]. However, Humbert's approach is restricted to exactly two participants. Giving each participant independent control over both axes allows the interface to be used by one or many individuals, expanding the range of possible applications.

Participants have the option to use eye blinks to trigger sounds and visual effects. Eye blinks are detected by calculating, in real-time, the standard deviation of the raw EEG signal within a short (200 ms) sliding window. If the standard deviation exceeds a predetermined threshold, a blink is indicated and an event may be triggered. For example, a blink can trigger a pulsatile flash which emphasizes the quadrant(s) wherein the participants' cursors are located, optionally accompanied by a short audio sample (e.g., a drum beat). Another option allows the use of predefined sequences of eye blinks within a specified time interval to trigger a larger repertoire of events, including additional audio samples or visual effects. Although some care is taken to reduce false positives, for instance, suppressing an event trigger if the interval between two consecutive suprathreshold events is less than 40 ms, as often occurs in movement or muscle artifacts, more complex and accurate blink detection routines can be implemented and may be used in future versions of the installation.

2.2 Mixing Music to Match Mental States

The music mix was designed to fulfill a few simultaneous expectations: first, each of the four tracks represents a combination of extreme values along the two axes, which, when listened to in isolation, were intended to subjectively represent that particular extreme state. For instance, one might include samples of beat-driven, yet ambient music to reflect an alert/focused yet relaxed state of mind. Alternately, more spastic, unpredictable samples of music could be chosen to reflect states of high focus and low relaxation (anxious, agitated). However, due to the aleatoric nature of the design, there is a relatively low probability of the tracks being heard on their own for a sustained period of time. The tracks mesh together effectively so that any combination of all four tracks would sound good together while also conveying a state in between the focus-relaxation extremes. This was primarily achieved in an intuitive fashion, by composing with music samples representing a particular extreme, but having a sonic palette and rhythmic profile in common.

3. DISCUSSION AND FUTURE WORK

Future manifestations of this system will focus on expanding the collaborative aspect of the design and explore other

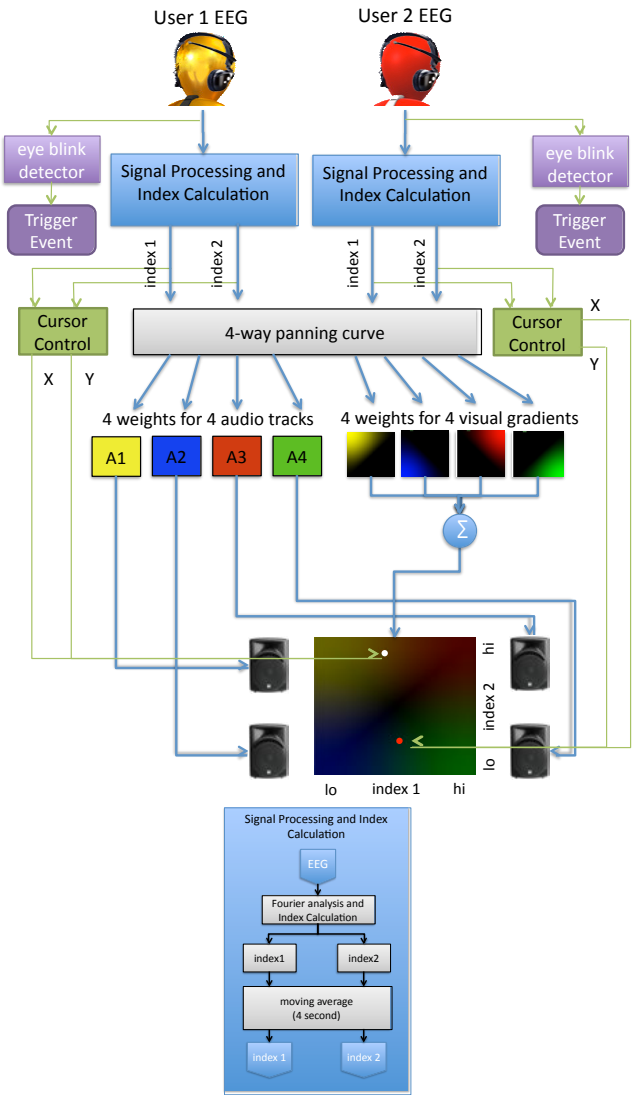


Figure 2: Diagram of the installation hardware setup with the communication protocols between components. In our installation, index1 corresponds to "relaxation/meditation" and index2 to "attention/focus"

treatments of EEG data to extract indices which can represent a wider range of cognitive and affective states. Due to its use of relatively inexpensive and easily obtainable hardware, *MoodMixer* can be readily extended to an arbitrary number of participants who can collaboratively control the system, linked over the Internet using the Open Sound Control (OSC) protocol. Conversely, *MoodMixer* can also currently be used by a single individual, with audio represented in either the four-channel surround sound configuration or collapsed to stereo for portable use.

Although the reported instantiation of *MoodMixer* uses Neurosky's measures of focus and relaxation, future instantiations may incorporate EEG-based measures of affective state, such as emotional arousal ("active" vs. "calm") and valence ("positive/good" vs. "negative/bad"). Recent research performed on human subjects listening to short music pieces [5, 4], has shown that changes in self-reported emotional arousal and valence induced by listening to music pieces, as well as tempo (fast/slow) and mode (major/minor) of listened music, is significantly correlated with

changes in EEG bandpower. Specifically, it was found that listening to minor mode music was correlated with increased frontal midline gamma power (25-60 Hz) while the converse was true for major mode music. Additionally, listening to slow-tempo music correlated with increased theta (4-8 Hz) activity. Furthermore, in a related study using the same music dataset, positive emotional valence was positively correlated with theta power and negatively correlated with delta power, while emotional arousal was positively correlated with both delta and theta power [5]. A recent study by our colleagues Onton and Makeig demonstrated that EEG features associated with 15 prototypical emotional states (Joy, Love, Frustration, ...), when mapped onto a two-dimensional plane using non-metric multidimensional scaling (MDS), formed a circumplex pattern with features corresponding to similar emotions located near each other in the MDS space, and negative valence emotions arrayed on the left and positive valence emotions on the right [10]. A natural extension of *MoodMixer* would allow participants to emotionally navigate a similar two-dimensional audiovisual space, wherein music tracks associated with each of a number of affective states are each assigned a corresponding coordinate in the transformed MDS space and continually mixed based on the participants' positions in the transformed MDS space. Future generations of the *MoodMixer* design may also incorporate algorithmic composition techniques to generate a musical "mood" mixture in real time which corresponds with the participants' affective or cognitive states.

Accurate detection of many of the aforementioned complex cognitive and affective states likely requires recording of signals from multiple brain regions. While our current instantiation of *MoodMixer* uses a single-electrode wearable EEG system, which is suitable for inferring basic states of arousal and cognitive load, the use of a multichannel system would provide data from more scalp locations allowing for calculation of interhemispheric differences as well as enabling robust spatiotemporal source separation techniques, such as Independent Component Analysis which can dramatically improve the signal to noise ratio and interpretability of the acquired data [7, 2]. This in turn would expand the range of cognitive and affective states that can be monitored for control of the system. Fortunately, wearable multichannel EEG hardware is now becoming ubiquitous with a number of established and nascent companies offering affordable multichannel "dry" (gel-free) electrode systems (Emotiv, Quasar, BrainProducts GmbH, PICO imaging, g.tec GmbH, to name a few). We are currently experimenting with a 16-channel wearable EEG system and plan to incorporate this technology into future instantiations of the installation.

Our use of wearable, wireless EEG technology introduces additional practical applications of *MoodMixer*. For instance, it may be used as a single-player or collaborative game in which players attempt to actively manipulate their cognitive or affective states to produce different musical/visual mixtures. Or it might be used as a computer desktop gadget/widget or mobile phone app in which an aesthetically-pleasing "background" electronic music mix (with optional visual texture) is continually generated based on passive monitoring of the user's mental state as they go about their day. Such a system may even have therapeutic applications, for example allowing one to inobtrusively monitor their own levels of stress or relaxation throughout the day. There are a number of possibilities, and we hope that this and other extensions of the *MoodMixer* concept will expand the existing repertoire of fun and aesthetically-pleasing systems for individual or collaborative, passive or active cognitive/affective sonification.

4. CONCLUSIONS

In this collaborative EEG sonification system, two participants control a music mix and corresponding visualization by actively or passively manipulating their cognitive state. This approach is novel in its collaborative design, and in that it affords both active and passive sonification approaches. *MoodMixer* represents a first step towards new media projects which explore new modes of social interaction and affective processing and control in brain-computer music interfaces.

5. ACKNOWLEDGMENTS

Thanks to Neurosky for donating two of their MindSets as part of their academic partnership program.

6. REFERENCES

[1] NeuroSky's eSense^{TM} Meters and Detection of Mental State. Technical report, 2009.

[2] A. Bell and T. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[3] M. Grierson. Composing With Brainwaves: Minimal Trial P300 Recognition As An Indication of Subjective Preference For the Control of a Musical Instrument. *International Computer Music Conference*, 2008.

[4] Y. Lin, J. Duann, J. Chen, and T. Jung. Electroencephalographic dynamics of musical emotion perception revealed by independent spectral components. *NeuroReport*, 21(6):410, 2010.

[5] Y. Lin, C. Wang, T. Jung, T. Wu, S. Jeng, J. Duann, and J. Chen. EEG-Based Emotion Recognition in Music Listening. *IEEE Transactions on Biomedical Engineering*, 57(7):1798–1806, 2010.

[6] A. Lucier. *Reflections: interviews, scores, writings= Reflexionen: Interviews, Notationen, Texte*. Köln: MusikTexte, 1995.

[7] S. Makeig, A. Bell, T. Jung, T. Sejnowski, et al. Independent component analysis of electroencephalographic data. *Advances in neural information processing systems*, pages 145–151, 1996.

[8] S. Mann, J. Fung, and A. Garten. DECONcert: Bathing in the light, sound, and waters of the musical brainbaths. 2007.

[9] E. Miranda and A. Brouse. Interfacing the Brain Directly with Musical Systems: On developing systems for making music with brain signals. *Leonardo*, 38(4):331–336, 2005.

[10] J. Onton and S. Makeig. High-frequency broadband modulations of electroencephalographic spectra. *Frontiers in human neuroscience*, 3, 2009.

[11] G. Rebolledo-Mendez, I. Dunwell, E. Martínez-Mirón, M. Vargas-Cerdán, S. de Freitas, F. Liarokapis, and A. García-Gaona. Assessing Neurosky's usability to detect attention levels in an assessment exercise. *New Trends in Human-Computer Interaction*, pages 149–158, 2009.

[12] D. Rosenboom. *Biofeedback and the arts: results of early experiments*. Aesthetic Research Centre of Canada, 1976.

[13] T. Zander, C. Kothe, S. Jatzev, and M. Gaertner. Enhancing human-computer interaction with input from active and passive brain-computer interfaces. *Brain-Computer Interfaces*, pages 181–199, 2010.

OSC Implementation and Evaluation of the Xsens MVN suit

Ståle A. Skogstad and Kristian Nymoen
fourMs group - Music, Mind, Motion, Machines
University of Oslo, Department of Informatics
{savskogs,krisny}@ifi.uio.no

Yago de Quay
University of Porto, Faculty of Engineering
Rua Dr. Roberto Frias, s/n
4200-465 Portugal
yagodequay@gmail.com

Alexander Refsum Jensenius
fourMs group - Music, Mind, Motion, Machines
University of Oslo, Department of Musicology
a.r.jensenius@imv.uio.no

ABSTRACT

The paper presents research about implementing a full body inertial motion capture system, the Xsens MVN suit, for musical interaction. Three different approaches for streaming real time and prerecorded motion capture data with Open Sound Control have been implemented. Furthermore, we present technical performance details and our experience with the motion capture system in realistic practice.

1. INTRODUCTION

Motion Capture, or MoCap, is a term used to describe the process of recording movement and translating it to the digital domain. It is used in several disciplines, especially for bio-mechanical studies in sports and health and for making lifelike natural animations in movies and computer games. There exist several technologies for motion capture [1]. The most accurate and fastest technology is probably the so-called infra-red optical marker based motion capture systems (IrMoCap)[11].

Inertial MoCap systems are based on sensors like accelerometers, gyroscopes and magnetometers, and perform *sensor fusion* to combine their output data to produce a more drift free position and orientation estimation. In our latest research we have used a commercially available full body inertial MoCap system, the Xsens MVN[1] suit [9]. This system is characterized by having a quick setup time and being portable, wireless, moderately unobtrusive, and, in our experience, a relatively robust system for on-stage performances. IrMoCap systems on the other hand have a higher resolution in both time an space, but lack these stage-friendly properties. See [2] for a comparison of Xsens MVN and an IrMoCap system for clinical gait analysis.

Our main research goal is to explore the control potential of human body movement in musical applications. New MoCap technologies and advanced computer systems bring new possibilities of how to connect human actions with musical expressions. We want to explore these possibilities and see how we can increase the connection between the human body's motion and musical expression; not only focusing on the performer, but also on how the audience perceives the performance.

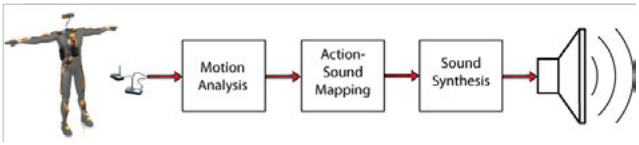


Figure 1: The Xsens suit and possible data flow when using it for musical interaction.

To our knowledge, we are among the first to use a *full body* inertial sensor based motion capture suit in a musical setting, and hence little related work exists. Lympouridis et. al. has used the inertial system Orient-2/-3 for sonification of *gestures* and created a framework for "bringing together dancers, composers and musicians" [6][5]. Meas et. al have used 5 inertial (Xsens) sensors to quantify the relation between sound stimuli and bodily response of subjects [7]. An upper body mechanical system has briefly been examined by [3]. See [11] for a review of related work in the area of IrMoCap for musical interaction.

In the next section, we will give a brief overview of the Xsens MVN technology. Then in section 3 we will report on three Open Sound Control implementations for the Xsens system and discuss some of our reflections. In section 4 we will give our evaluation and experience with the Xsens MVN system, before we propose a technology independent real time MoCap toolbox in section 5.

2. THE XSENS MVN TECHNOLOGY

The Xsens MVN technology can be divided into two parts. First, the sensor and communication hardware are responsible for collecting and transmitting the raw sensor data. Second, these data are treated by the Xsens MVN software engine, which interprets and reconstructs the data to full body motion while trying to minimize drift.

2.1 The Xsens MVN Suit (Hardware)

The Xsens MVN suit consists of 17 inertial MTx sensors, which are attached to key areas of the human body [9]. Each sensor consists of a 3D gyroscope, 3D accelerometer and magnetometer. The raw signals from the sensors are connected to a pair of Bluetooth 2.0 based wireless transmitters, which transmit the raw motion capture data to a pair of wireless receivers. The total weight of the suit is approximately 1.9 kg and the whole system comes in a suitcase with the total weight of 11 kg.

2.2 The Xsens MVN engine (Software)

The data from the Xsens MVN suit is fed to the MVN software engine that uses sensor fusion algorithms to produce

[1] *Xsens MVN* (MVN is a name not an abbreviation) is a motion capture system designed for the human body and is not a generic motion capture device.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

absolute orientation values, which are used to transform the 3D linear accelerations to global coordinates. These in turn are translated to a human body model which implements joint constraints to minimize integration drift [9].

The Xsens MVN system outputs information about body motion by expressing body postures sampled at a rate up to 120Hz. The postures are modelled by 23 body segments interconnected with 22 joints [9]. The Xsens company offers two possibilities of using the MVN fusion engine: the Windows based *Xsens MVN Studio* and a software development kit called *Xsens MVN SDK*.

2.3 How to use the System

There are three main suit configurations; full body, upper body or lower body. When the suit is properly configured, calibration is needed to initialize the position and orientation of the different body segments. When we are satisfied with the calibration the system can be used to stream the motion data to other applications in real-time or perform recordings for later playback and analysis.

How precise one needs to perform the calibration may vary. We have found that so-called *N-pose* and *T-pose* calibrations are the most important. A *hand touch* calibration is recommended if a good relative position performance between the left and right hand is wanted. Recalibration can be necessary when the system is used over a longer period of time. It is also possible to input body measurements of the tracked subject to the MVN engine, but we have not investigated if this extra calibration step improves the quality of data for our use.

In our experience, setting up the system can easily be done in less than 15 minutes compared to several hours for IrMoCap systems [2].

2.4 Xsens MVN for Musical Interaction

A typical model for using the Xsens suit for musical application is shown in Figure 1. In most cases, motion data from the Xsens system must be processed before it can be used as control data for the sound engine. The complexity of this stage can vary from simple scaling of position data to more complex pattern recognition algorithms that look for mid/higher-level cues in the data. We will refer to this stage as *cooking* the motion capture data.

The main challenges of using the Xsens suit for musical interaction fall into two interconnected groups. Firstly, the purely technical challenges, such as minimizing latency, managing network protocols and handling data. Secondly, the more artistic challenges involving questions like how to make an aesthetically pleasing connection between action and sound. This paper will mainly cover the technical challenges.

3. IMPLEMENTATION

To be able to use the Xsens MVN system for musical interaction, we need a way to communicate the data that the system senses to our musical applications. It was natural to implement the OSC standard since the Xsens MVN system offers motion data which is not easily related to MIDI signals. OSC messages are also potentially easier to interpret since these can be written in a human readable form.

3.1 Latency and Architecture Consideration

Low and stable latency is an important concern for *real-time* musical control [12]. This is therefore an important issue to consider when designing our system. Unfortunately, running software and sending OSC messages over normal computer networks offers inadequate support for synchronization mechanisms, since standard operating systems do not support this without dedicated hardware [10]. In our experience, to get low latency from the Xsens system, the software needs to run on a fast computer that is not overloaded with other demanding tasks. But how can we further minimize the latency?

3.1.1 Distribution of the Computational Load

From Figure 1 we can identify three main computationally demanding tasks that the data need to traverse before ending up as sound. If these tasks are especially demanding, it may be beneficial to distribute these computational loads to different computers. In this way we can prevent a computer from suffering too much from computational load, which can lead to a dramatic increase of latency and jitter. This is possible with fast network links and a software architecture that supports the distribution of computational loads. However, it comes at the cost of extra network overhead, so one needs to check if the extra cost does not exceed the benefits.

3.1.2 The Needed Communication Bandwidth

The amount of data sent through a network will partly be related to the experienced network latency. For instance, we should try to keep the size of the OSC bundles lower than the maximum network buffer size,[2] if the lowest possible network latency is wanted. If not, the bundle will be divided into several packages [10]. To achieve this, it is necessary to restrict the amount of data sent. If a large variety of data is needed, we can create a dynamic system that turns different data streams on when needed.

3.2 OSC Implementations

There are two options for using the Xsens MVN motion data in real time, either we can use the Xsens Studio's UDP network stream, or make a dedicated application with the SDK. The implementation must also support a way to effectively cook the data. We begun using the UDP network stream since this approach was the easiest way to start using the system.

3.2.1 MVN Network Stream Unpacker in Max/MSP

A MXJ Java datagram unpacker was made for Max/MSP, but the implementation was shown to be too slow for real time applications. Though a dedicated Max external (in C++) would probably be faster, this architecture was not chosen for further development since Max/MSP does not, in our opinion, offer an effective data cooking environment.

3.2.2 Standalone Datagram Unpacker and Cooker

We wanted to continue using the Xsens Studio's UDP network stream, but with a more powerful data cooking environment. This was accomplished by implementing a standalone UDP datagram unpacking application. The programming language C++ was chosen since this is a fast and powerful computational environment. With this implementation we can either cook the data with self produced code or available libraries. Both raw and cooked data can then be sent as OSC messages for further cooking elsewhere or to the final sound engine.

3.2.3 Xsens MVN SDK Implementation

The Xsens MVN software development kit offers more data directly from the MVN engine compared to the UDP network stream. In addition to position, we get: positional and angular acceleration, positional and angular velocity and information about the sensor's magnetic disturbance. Every

[2]Most Ethernet network cards support 1500 bytes. Those supporting Jumbo frames can support up to 9000 bytes.

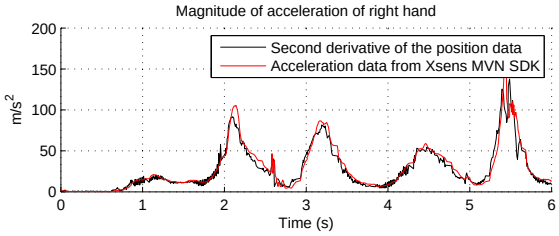


Figure 2: Difference between the second derivative of the position data versus the acceleration data obtained directly from MVN engine (SDK).

time frame is also marked with a time stamp that can be useful for analysis and synchronizing. Another benefit is that we have more control since we are directly communicating with the MVN engine and not listening for UDP packages. The drawback with the SDK is that we lose the benefit of using the user friendly MVN Studio and its GUI.

We implemented a terminal application with the SDK, that supports the basic Xsens features (calibration, playback, etc.). Since the application is getting data directly from the MVN engine we can save network overhead by cooking them in the same application before sending them as OSC messages. We also implemented a function that can send the motion data in the same data format as the Network UDP Datagram stream. This stream can then be opened by MVN Studio to get real-time visual feedback of the MoCap data.

3.2.4 Discussion

Since the solution presented in 3.2.2 offered a fast environment for data cooking, and let us use the user friendly MVN Studio, we have mainly used this approach in our work. We later discovered that the network stream offered by MVN Studio suffers from frame loss when driven in live mode, which affects both solutions presented in 3.2.1 and 3.2.2. Because of this we plan to focus on our SDK implementation in the future. An added advantage is that we no longer need to differentiate the segments positional data to be able to get properties like velocity and acceleration, since the SDK offers this directly from the MVN Engine. These data, especially the acceleration, seems to be of a higher quality since they are computed directly on the basis of the Xsens sensors and not differentiated from estimated position data as shown in Figure 2.[3]

3.3 Cooking Full Body MoCap Data

The Xsens MVN offers a wide range of different data to our system. If we use the network stream from the MVN Studio, each frame contains information about the position and orientation of 23 body segments. This yields in total 138 floating points numbers at a rate of 120Hz. Even more data will be available if one instead uses the MVN SDK as the source. Also different transformations and combinations of the data can be of interest, such as calculating distances or angles between body limbs.

Furthermore, we can differentiate all the above mentioned data to get properties like velocity, acceleration and jerk. Also, filters can be implemented to get smoother data or to emphasize certain properties. In addition, features like quantity of motion or "energy" can be computed. And with pattern recognition techniques we have the potential to recognize even higher level features [8].

We are currently investigating the possibilities that the Xsens MVN suit provides for musical interaction, but the mapping discussion is out of scope for this paper. Nevertheless, we believe it is important to be aware of the characteristics of the data we are basing our action-sound mappings on. We will therefore present technical performance details of the Xsens MVN system in the following section.

[3]The systems that tries to minimize positional drift probably contributes to a mismatch between differentiated positional data and the velocity and acceleration data from the MVN engine.

4. PERFORMANCE

4.1 Latency in a Sound Producing Setup

To be able to measure the typical expected latency in a setup like that of Figure 1 we performed a simple experiment with an audio recorder. One laptop was running our SDK implementation and sent OSC messages containing the acceleration of the hands. A patch in Max/MSP was made that would trigger a simple impulse response if the hands' acceleration had a high peak, which is a typical sign of two hands colliding to a sudden stop. The time difference between the acoustic hand clap and the triggered sound should then indicate the typical expected latency for the setup.

The Max/MSP patch was in experiment 1 running on the same laptop[4] as the SDK. In experiment 2 the patch was run on a separate Mac laptop[5] and received OSC messages through a direct Gbit Ethernet link. Experiment 3 was identical to 2 except that the Mac was replaced with a similar Windows based laptop. All experiments used the same firewire soundcard, *Edirol FA-101*. The results are given in Table 1 and are based on 30 measurements each which was manually examined in audio software. The standard deviation is included as an indication of the jitter performance. We can conclude that experiment 2 has the fastest sound output response while experiments 1 and 3 indicate that the Ethernet link did not contribute to a large amount of latency.

The Xsens MVN system offers a direct USB connection as an option for the Bluetooth wireless link. We used this option in experiment 4, which was in other ways identical to experiment 2. The results indicate that the direct USB connection is around 10-15 milliseconds faster and has a lower jitter performance than the Bluetooth link.

The upper boundary for "intimate control" has been suggested to be 10ms for latency and 1ms for its variations (jitter) [12]. If we compare the boundary with our results, we see that overall latencies are too large and that the jitter performance is even worse. However, in our experience, the system is still usable in many cases dependent on the designed action-sound mappings.

Table 1: Statistical results of the measured action to sound latency, in milliseconds.

Experiment	min	mean	max	std. dev.
1 Same Win laptop	54	66.7	107	12.8
2 OSC to Mac	41	52.2	83	8.4
3 OSC to Win	56	68	105	9.8
4 OSC to Mac - USB	28	37.2	56	6.9

4.2 Frame Loss in the Network Stream

We discovered that the Xsens MVN Studio's (version 2.6 and 3.0) network stream is not able to send all frames when running at 120Hz in real time mode on our computer.[3] At this rate it is skipping 10 to 40 percent of the frames. This does not need to be a significant problem if one use "time independent" analysis, that is analysis that does not look at the history of the data. But if we perform differential calculations on the Xsens data streams, there will be large jumps

[4]Dell Windows 7.0 Intel i5 based laptop with 4GB RAM
[5]MacBook Pro 10.6.6, 2.66 GHz Duo with 4GB RAM

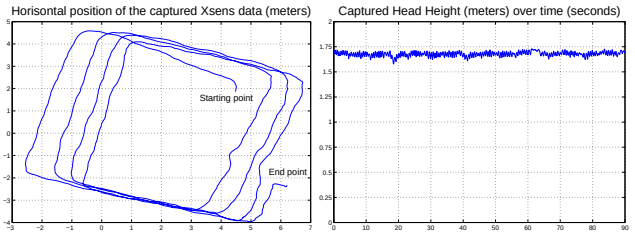


Figure 3: Plots of the captured horizontal (left) and vertical (right) position of the head.

in differentiated values during lost frames, hence noise. This was partly dealt with in the implementation described in 3.2.2. Whenever frames are detected as missing, the software will perform an interpolation. However, frame loss is still a major problem since we are not getting all the motion capture data and can lose important details in the data stream. For instance, if a trigger algorithm is listening for some sudden action, a couple of lost frames can make the event unrecognisable.

4.3 Positional Drift

The sensors in the Xsens MVN suit can only observe relative motion and calculate position through integration. This introduces drift. To be able to observe this drift we conducted a simple test by letting a subject walk along a rectangular path (around 6x7 meters) four times. Figure 3 shows a horizontal positional drift of about 2 meters during the 90 second long capture session. We can therefore conclude that Xsens MVN is not an ideal MoCap system if absolute horizontal position is needed.[6] The lack of drift in the vertical direction however, as can be seen in the right plot in Figure 3, is expected since the MVN engine maps the data to a human body model and assumes a fixed floor level.

4.4 Floor Level

If the motion capture area consists of different floor levels, like small elevated areas, the MVN engine will match the sensed raw data from the suit against the floor height where the suit was calibrated. This can be adjusted for in the post processing, but the real-time data will suffer from artifacts during floor level changes.

4.5 Magnetic Disturbance

The magnetic disturbance is critical during the calibration process but does not, to our experience, alter the motion tracking quality dramatically. During a concert we experienced significant magnetic disturbance, probably because of the large amount of electrical equipment on stage. But this did not influence the quality of MoCap data in such a way that it altered our performance.

4.6 Wireless Link Performance

Xsens specifies a maximum range up to 150 meters in an open field [13]. In our experience the wireless connection can easily cover an area with a radius of more than 50 meters in open air. Such a large area cannot be practically covered using IrMoCap systems.

We have performed concerts in three different venues.[7] During the two first concerts we experienced no problems with the wireless connection. During the third performance we wanted to test the wireless connection by increasing the distance between the Xsens suit and the receivers to about 20 meters. The wireless link also had an added challenge since the concert was held in a conference venue where we expected constant WIFI traffic. This setup resulted in problems with the connection and added latency. The distance should therefore probably be minimized when performing in venues with considerable wireless radio traffic.

[6]The product *MVN MotionGrid* will improve this drift.

[7]First concert: `www.youtube.com/watch?v=m1OffxIArrAi`

4.7 Final Performance Discussion

We believe that the Xsens MVN suit, in spite of its shortcomings in latency, jitter and positional drift, offers useful data quality for musical settings. However, the reported performance issues should be taken into account when designing action-sound couplings. We have not been able to determine whether the Xsens MVN system preserves the motion qualities we are most interested in compared to other MoCap systems, nor how their performance compares in real life settings. To be able to answer more of these questions we are planning systematic experiments comparing Xsens MVN with other MoCap technologies.

5. FUTURE WORK

In Section 3.3 we briefly mentioned the vast amount of data that is available for action-sound mappings. Not only are there many possibilities to investigate, it also involves many mathematical and computational details. However, the challenges associated with the cooking of full body MoCap data are not specific to the Xsens MVN system. Other motion capture systems like IrMoCap systems offer similar data. It should therefore be profitable to make one cooking system that can be used for several MoCap technologies.

The main idea is to gather effective and fast code for real time analysis of motion capture data; not only algorithms but also knowledge and experience about how to use them. Our implementation is currently specialized for the the Xsens MVN suit. Future research includes incorporating this implementation with other motion capture technologies and develop a real time motion capture toolbox.

6. REFERENCES

[1] http://en.wikipedia.org/wiki/motion_capture.

[2] T. Cloete and C. Scheffer. Benchmarking of a full-body inertial motion capture system for clinical gait analysis. In *EMBS*, pages 4579 –4582, 2008.

[3] N. Collins, C. Kiefer, Z. Patoli, and M. White. Musical exoskeletons: Experiments with a motion capture suit. In *NIME*, 2010.

[4] R. Dannenberg. *Real-time scheduling and computer accompaniment*. MIT Press, 1989.

[5] V. Lympourides, D. K. Arvind, and M. Parker. Fully wireless, full body 3-d motion capture for improvisational performances. In *CHI*, 2009.

[6] V. Lympouridi, M. Parker, A. Young, and D. Arvind. Sonification of gestures using specknets. In *SMC*, 2007.

[7] P.-J. Maes, M. Leman, M. Lesaffre, M. Demey, and D. Moelants. From expressive gesture to sound. *Journal on Multimodal User Interfaces*, 3:67–78, 2010.

[8] G. Qian, F. Guo, T. Ingalls, L. Olson, J. James, and T. Rikakis. A gesture-driven multimodal interactive dance system. In *ICME*, 2004.

[9] D. Rosenberg, H. Luinge, and P. Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. *Xsens Technologies*, 2009.

[10] A. Schmeder, A. Freed, and D. Wessel. Best practices for open sound control. In *LAC*, 2010.

[11] S. Skogstad, A. R. Jensenius, and K. Nymoen. Using ir optical marker based motion capture for exploring musical interaction. In *NIME*, 2010.

[12] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. In *NIME*, 2001.

[13] Xsens Technologies B.V. *Xsens MVN User Manual*.

The effect of visualizing audio targets in a musical listening and performance task

Lonce Wyse[1, 2] Norikazu Mitani[2] Suranga Nanayakkara[2]

[1] Department of Communications and New Media
[2] Arts and Creativity Laboratory
National University of Singapore
{lonce.wyse, norikazu.mitani, suranga}@nus.edu.sg

ABSTRACT
The goal of our research is to find ways of supporting and encouraging musical behavior by non-musicians in shared public performance environments. Previous studies indicated simultaneous music listening and performance is difficult for non-musicians, and that visual support for the task might be helpful. This paper presents results from a preliminary user study conducted to evaluate the effect of visual feedback on a musical tracking task. Participants generated a musical signal by manipulating a hand-held device with two dimensions of control over two parameters, pitch and density of note events, and were given the task of following a target pattern as closely as possible. The target pattern was a machine-generated musical signal comprising of variation over the same two parameters. Visual feedback provided participants with information about the control parameters of the musical signal generated by the machine. We measured the task performance under different visual feedback strategies. Results show that single parameter visualizations tend to improve the tracking performance with respect to the visualized parameter, but not the non-visualized parameter. Visualizing two independent parameters simultaneously decreases performance in both dimensions.

Keywords
Mobile phone, Interactive music performance, Listening, Group music play, Visual support

1. INTRODUCTION
The last ten years or so has seen the development of many novel interactive media devices designed to support the engagement of participants through sound. Many are explicitly designed for anyone to enjoy, not only those with specific musical skills. Examples include the hyperinstruments created for the Brain Opera [15], and a variety of different table top devices such as Jam-O-Drum [1], and Reactable [10], "new media art" installations that involve sound, and musical games. Recent developments in mobile phones are also inspiring new kinds of interactive music [14,18]. Many of these devices and systems offer the potential for collective sound play that we might recognize as a form of musical improvisation. Recently, sensor-rich mobile phones have become ubiquitous computational devices providing new opportunities as public interfaces for media and musical performance environments.
Because of their computational and communicative power and their ubiquity, mobile phones hold enormous potential for supporting essentially an unlimited number of people to participate in interactive musical environments [16]. However there are still barriers to spontaneous and rewarding musical engagement for many due to a lack of musical experience.
Professional musicians exhibit a wide variety of sophisticated improvisational behaviors including conversational patterns, complimentary role-playing, coordinated transitions, etc. much more than non-musicians do. These patterns of improvisatory play may exploit technical skills, but don't seem to depend critically on their complexity. Improvisation is rather a skill that depends on the ability to synchronize one's own physical actions while simultaneously maintaining an awareness of another's musical activity [3]. Considerable attention has been devoted to providing non-musicians with instrumental interface that sounds musical without the need for technical skills [12], but less to supporting non-musicians in collaborative improvisation without having previous training in the requisite musical listening and communications skills, although there are exceptions [5][2].
Using the rotational dimensions of hand-held devices as simple instrument interfaces, we have been exploring whether and how graphical displays can be used to support non-musicians in the kind of listening and performance practices that make for engaging collaborative improvisational behavior. A preliminary study using mobile phones for collaborative music making showed that non-musicians often get lost in their attempt to listen to others while they concurrently engaged in playing their own instrument. Previous music cognition studies [11] show difference in listening skills of professional, amateur, and non-musicians. It has also been shown that non-musicians do not perform gestural imitation tasks as well as musicians [17].
The challenges faced by non-musicians in simultaneous music listening and performance motivates our exploration of visual feedback to support and encourage musical behavior for the musically untrained. This paper reports on a user study we conducted on the effect of different kinds of visual feedback to guide behavior.

2. RELATED WORK
It is known that real-time feedback is a useful tool for teaching singers to sing on pitch [8]. While these results are primarily about self-monitoring, target pitches are generally simultaneously visualized [9]. These systems are also oriented toward the long-term effects of learning, not just the performance during learning itself. Some of the visualizations described use multiple windows showing two parameters at once (pitch and spectral information), but there is little in this literature about the attentional issues of multiple displays of independent information. Wilson [19] found that computer-based visual feedback helped singers increase pitch accuracy, but showed differences between results for novice and advance singer. This result shows that supporting novices presents

NIME'11, 30 May–1 June 2011, Oslo, Norway.

different challenges than supporting experts due to the different musical skill levels.

Music has many simultaneous dimensions changing at the same time that non-musicians in particular may have trouble tracking. Our experimental task requires that participants pay attention to two auditory features simultaneously. Previous work on auditory feature conjunction has focused mainly on the detection of specific stimuli in the presence of auditory distracters [20]. Although our task is not primarily one of detection, it is based on the idea that features can be separately attended, and that attention can be modulated with visual displays.

In our experiments, features of the visual stimulus are co-modulated with auditory features creating an audio-visual object akin to audio-visual speech integration that might direct attention to some auditory features at the expense of others that are not co-modulated with visual features. Intersensory facilitation is well established as decreasing reaction times [7]. Visual speech cues matched to auditory speech can enhance the detection of speech in noise [6]. It is also possible that dual-modality presentations reduce cognitive load thereby facilitating performance improvement in learning tasks [13].

The effect of visual feedback on music performance has also studied by Brandmeyer et al. [4]. A user study was conducted comparing two different visual feedback strategies for supporting the imitation of recorded material. One strategy was to show high-level abstract feedback reflecting expressive styles of performance. Another showed low-level descriptive feedback such as timing and dynamics of individual notes. Their results showed that for musicians, the more abstract high-level visual feedback improved imitation performance better than the low-level descriptive feedback.

3. HYPOTHESES

Based on previous literature and our preliminary studies, indications are that it may be possible to improve the performance of non-musicians in simultaneous musical listening and performance tasks with the support of visual feedback. In the present study, a sequence of tones is generated as a target pattern by the computer with two dimensions of variation: musical pitch, and density of note events. The subject uses a mobile phone as an interface with the same dimensions of control, and is given the task of tracking the target pattern as closely as possible. The two musical dimensions change smoothly with the rotational angle of the mobile phone so that there is no need for tightly coordinated temporal gestures (such as "hitting" a note). This makes physically playing the instrument easy and allows for attention to be directed toward listening and task execution.

The task was performed under different visualization conditions that provided information about neither, one, or both of the musical dimensions of the computer-generated pattern. The tracking task was the same under all visualization conditions, and optimal task performance still always required listening.

We report here on the results of testing two hypotheses:

H1. Visual feedback about the target pattern would result in better performance than no visual feedback condition.

H2. Visual feedback of one machine target pattern dimension would improve the participants' performance in that dimension only.

4. USER STUDY

Twenty-eight participants (two male participants and 26 female participants) took part in the study. Their median age was 21 years ranging from 20 to 25 years. Eight participants had studied music as a subject in primary and secondary school level and eleven of them were entirely inexperienced with musical instruments although they reported a wide range of musical listening tastes. The participants were recruited from the university student community. Participants were given a mobile phone with embedded accelerometers running an application to control sounds by rotating the phone up/down and left/right. They were asked to follow a machine generated sound pattern under different visual feedback conditions. The study was conducted in accordance with the ethical research guidelines provided by the Internal Review Board (IRB) of the National University of Singapore and with IRB approval.

4.1 Apparatus

The study was carried out in a quiet room with a 42 -inch LCD display, two speakers and office chair where participants were seated during the study. The visual display was placed at a constant horizontal distance (approximately 240 cm) from the chair and constant elevation (approximately 100 cm) from the floor. Two speakers were places on each side of the screen to present audio feedback.

The participant's mobile phone interface controlled the sound of an acoustic piano with two parameters. Rotating the mobile phone up and down changed the pitch of notes from A5 (MIDI note #81) to C3 (MIDI note #48) along a pentatonic scale. Rotating the mobile phone left and right changed the density of note events from slow to fast (143 BPM to 900 BPM). The effective angles were within 45 degrees (up/down or left/right) from center (keeping the phone parallel to ground and pointing directly towards display screen). Sound was toggled on and off with a tap of the thumb on the touch screen of the mobile phone.

Sound controlled by the participant was played from the speaker on the left-hand side of the screen. The computer-generated pattern was made with the sound of a marimba, and was played from the speaker on the right-hand side of the screen. The assignment of different instrument timbres (piano and marimba), and different stereo channels for playing the two source patterns was designed to make it easy for users to discriminate between the sounds patterns generated by the computer and themselves. A follow-up questionnaire confirmed that none of the participants had difficulty differentiating between the two simultaneous instrument sources.

Parameters for the pitch and the density-of-note events for the computer pattern were read from the same look-up table for each session to insure that the difficulty of the tracking task for each participant was exactly the same, but they were initiated from random starting points in the tables to minimize the possibility that patterns could be learned. The system outline is shown in Figure 1.

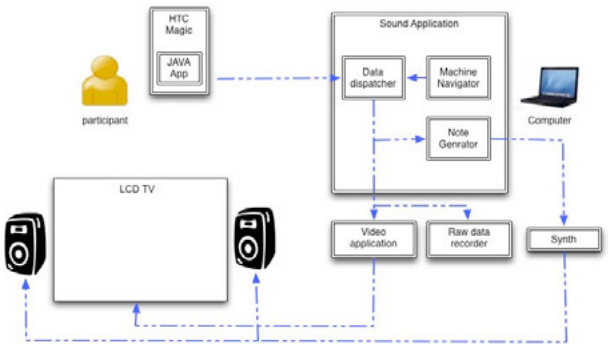


Figure 1. System outline

4.2 Procedure

In each trial, both participants and the computer played their instruments simultaneously for a period of 45 seconds during which time the participant engaged in the task of following the musical pattern of the computer as closely as possible.

Each of the trials was accompanied by visualizations tracking different control parameters generated by the subject and/or the computer instrument except for one segment where there was no visual feedback. Participants were informed about which controls would be visually tracked at the beginning of each trial. In addition, a color-coded legend appeared in the right hand side of the display to highlight what was being visualized. Visualizations mapped parameters in the vertical dimension (high pitch and high density were plotted higher in the vertical axis), and the graph scrolled to show a history of the parameter value (see Figure 2).

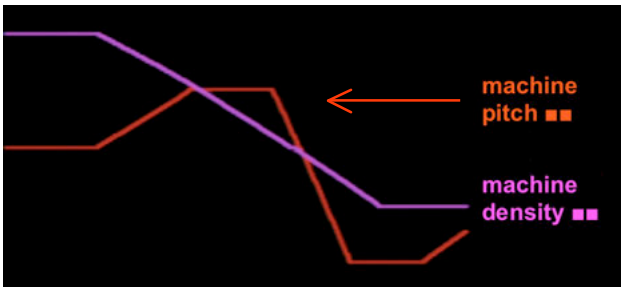


Figure 2. A typical visualization tracking both pitch and density of events (PD condition). Real-time measurements appear on the right as the visualization scrolls to the left to show a history of the parameter values. [Note: Arrow pointer was not a part of the visualization; but has been added for black and white viewing]

We tested ten conditions in total, but here we will consider only (a) visualization tracking the pitch of the computer's sound, P; (b) visualization tracking density of event of the computer's sound, D; (c) visualization that tacks both pitch and density of events of the computer's sound, PD; (d) no visual feedback, N; to address the hypothesis presented in this paper.

Before starting the study, each participant was told that the purpose of the experiment was to study the relationship between visualizations and making sound with a hand-held instrument. In addition, they were given the chance to become comfortable with the mobile phone instrument. They were also shown four examples of visualizations tracking pitch or density of events of their own instrument and pitch or density of the computer generated pattern. Once the participant was ready, they were reminded that their task was to follow the sound pattern of the computer as closely as possible. Trials were presented in a random order across subjects.

Data from the movement of the phone in the two control dimensions as well as the parameters generating the target pattern were recorded in a log file. After all the trials, the participants were asked to share their experience by answering a questionnaire. Each subject took approximately 30 minutes to complete the experiment session. It took two days to collect responses from 28 participants.

4.3 Analysis

Data was analyzed by comparing the two dimensional movement of the phone by the participant with the parameters controlling the synthesized target pattern. Data collected from four participants were discarded from the analysis since had accidently toggled off the sound during some of the trials. In each trial, it took a few seconds for the participants to get used to the task of following the computer-generated sound. Therefore, a 30 second interval from 13th second to 43rd second was chosen to evaluate the performance of the sound-following task. Pitch and density parameter values from both the participant and computer controlled patterns were sampled at 25 samples per second to calculate three performance indicators as follows:

Error in pitch dimension, $E_p = \sum_{t=13}^{43} \frac{|p_u - p_m|}{n}$

Error in density dimension, $E_d = \sum_{t=13}^{43} \frac{|d_u - d_m|}{n}$

Total error, $E_t = \sum_{t=13}^{43} \sqrt{E_p^2 + E_d^2}$

Where p_u is user's pitch parameter, p_m is computer's pitch parameter, d_u is user's pitch parameter, d_m is computer's pitch parameter, and n is the number of number of data points. Figure 3 shows the error rates across all experimental conditions.

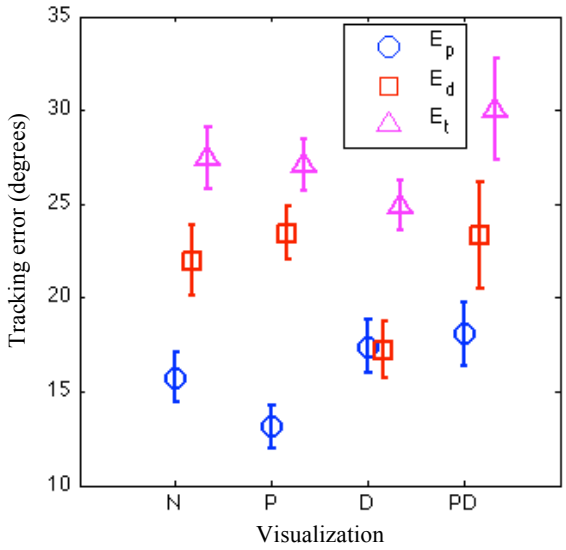


Figure 3: Plot of error rates in the pitch dimension (Ep), density dimension (Ed) and total error (Et) with 95% confidence intervals for four different experimental conditions [N–no visual feedback, P–visualization tracking computer's pitch, D–visualization tracking computer's density of events, Visualization tracking both P and D]

As seen in Figure 3, error in the pitch dimension is at its lowest when the visualization tracks the pitch of the computer sound. Similarly, error in the density dimension is lowest when the visualization tracks the density of events of the computer pattern. These results support the hypothesis that visualization of a target pattern control parameter reduces the tracking error in that dimension.

In general, error in the density dimension is significantly higher than the error in pitch dimension except in condition D, where the visualization tracks density parameter of the computer sound. However, this bias does not affect the cross-condition comparisons upon which we base the evaluation of our hypotheses.

Compared to the no visualization condition (D), total error in the tracking performance is slightly reduced when a single parameter is visualized (conditions P and D), however the improvement in total performance is less than the improvement in the visualized parameter. In both single-parameter tracking cases (P and D), the performance on the non-visualized

parameter was slightly worse than in the no visualization condition (N). It would appear that the increased performance in the visualized parameter is at the expense of some degree of performance in the non-visualized parameter.

Given the improvement in performance for each of the individual parameter visualization conditions, one might predict that visualizing both parameters would result in a further performance improvement. However, when both parameters were visualized (condition PD), total performance error was significantly greater than all other conditions. Participants were sensitive to the difficulty of the task in this condition, and reported in the questionnaire following the experiment that trying to keep track of two visual parameters distracted them from the task of following the sound of the computer.

5. CONCLUSION AND FUTURE WORK

This study explored the effect of providing visual feedback with information about parameters of a target audio pattern of control parameters on a music-following task. We developed a mobile phone based interface to be simple to understand and play without any training. The results support both our hypotheses (*H1* and *H2*) concerning performance with single-parameter target visualization.

The data and the responses to the questionnaire following the experiment show that the density parameter was harder to follow than pitch. However, it is impossible to draw conclusions about any inherent difference in tracking these parameters outside of the specific ranges and mapping strategies used in our experiments. We expect that the particular characteristics of the mapping and visualization strategies have a significant effect on performance, and this warrants further study. Finally, our present study focused only on showing target parameters, although it seems likely that visual feedback of user activity would also influence listening and performance behavior.

6. ACKNOWLEDGEMENTS

We would like to express our gratitude to Angela Khoo for her input on aspects of the experimental design. This work was supported by project grant NRF2007IDM-IDM002-069 from the Interactive and Digital Media Project Office, Media Development Authority, Singapore, and the NUS AcRF project, “Listening Strategies for New Media; Experience and Expectation”.

7. REFERENCES

[1] Blaine, T. and Perkis, T. The Jam-O-Drum interactive music system: a study in interaction design. *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, (2000), 165-173.

[2] Blaine, T. and Fels, S. Contexts of Collaborative Musical Experiences. *Proceedings of the International Conference on New Interfaces for Musical Expression*, (2003).

[3] Borgo, D. and Goguen, Sync or Swarm: Group Dynamics in Musical Free Improvisation. In R. Parncutt, A. Kessler, and F. Zimmer (eds.), Proceedings, Conference on Interdisciplinary Musicology, pages 52-53. Dept. Musicology, Graz (2004).

[4] Brandmeyer, A., Timmers, R., Sadakata, M., and Desain, P. Learning expressive percussion performance under different visual feedback conditions. *Psychological Research*, (2010).

[5] Weinberg, G. and Driscoll, S. *iltur:* Connecting Novices and Experts Through Collaborative Improvisation, *Proceedings of the International Conference on New Interfaces for Musical Expression* (2005), 17-22.

[6] Grant, K.W. and Seitz, P.F. The use of visible speech cues for improving auditory detection of spoken sentences. *The Journal of the Acoustical Society of America 108*, (2000), 1197.

[7] Hershenson, M. Reaction time as a measure of intersensory facilitation. *Journal of Experimental Psychology 63*, 3 (1962), 289-293.

[8] Hoppe, D., Sadakata, M., and Desain, P. Development of real-time visual feedback assistance in singing training: a review. *Journal of Computer Assisted Learning 22*, 4 (2006), 308-316.

[9] Howard, D.M. and Welch, G.F. Visual displays for the assessment of vocal pitch matching development. *Applied Acoustics 39*, 4 (1993), 235-252.

[10] Jordà, S., Geiger, G., Alonso, M., and Kaltenbrunner, M. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, (2007), 139-146.

[11] Kreutz, G., Schubert, E., and Mitchell, L.A. Cognitive Styles of Music Listening. *Music Perception: An Interdisciplinary Journal 26*, 1 (2008), 57-73.

[12] Machover, T. Shaping minds musically. *BT Technology Journal 22*, 4 (2004), 171-179.

[13] Mousavi, S.Y., Low, R., Sweller, J. Reducing cognitive load by mixing auditory and visual presentation modes. *Journal of Educational Psychology 87*, 2 (1995), 319-334.

[14] Oh, J., Herrera, J., Bryan, N.J., Dahl, L., and Wang, G. Evolving The Mobile Phone Orchestra. *Proceedings of the International Conference on New Interfaces for Musical Expression*, (2010).

[15] Paradiso, J. The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance. *Journal of New Music Research 28*, 2 (1999), 130-149.

[16] Scheible, J. and Ojala, T. MobiLenin combining a multi-track music video, personal mobile phones and a public display into multi-user interactive entertainment. *Proceedings of the 13th annual ACM Iinternational Conference on Multimedia, November*, (2005), 6-11.

[17] Spilka, M.J., Steele, C.J., and Penhune, V.B. Gesture imitation in musicians and non-musicians. *Experimental Brain Research 204*, 4 (2010), 549-558.

[18] Weinberg, G., Beck, A., and Godfrey, M. ZooZBeat: a Gesture-based Mobile Music Studio. *Proceedings of the 9th International Conference on New Interfaces of Musical Expression*, (2009), 312-315.

[19] Wilson, P.H., Lee, K., Callaghan, J., and Thorpe, C.W. Learning to sing in tune: Does real-time visual feedback help? *CIM07: 3rd Conference on Interdisciplinary Musicology,* Tallinn, Estonia, (2007), 15-19.

[20] Woods, D.L. and Alain, C. Conjoining three auditory features: an event-related brain potential study. *Journal of Cognitive Neuroscience 13*, 4 (2001), 492-509.

Composability for Musical Gesture Signal Processing using new OSC-based Object and Functional Programming Extensions to Max/MSP

Adrian Freed
CNMAT
Dept. of Music
UC Berkeley
adrian@cnmat.berkeley.edu

John MacCallum
CNMAT
1750 Arch Street
Berkeley, CA 94709
johnmac@berkeley.edu

Andy Schmeder
CNMAT
1750 Arch Street
Berkeley, CA 94709
schmeder@berkeley.edu

ABSTRACT
An effective programming style for gesture signal processing is described using a new library that brings efficient run-time polymorphism, functional and instance-based object-oriented programming to Max/MSP. By introducing better support for generic programming and composability Max/MSP becomes a more productive environment for managing the growing scale and complexity of gesture sensing systems for musical instruments and interactive installations.

Keywords
Composability, object, Open Sound Control, Gesture Signal Processing, Max/MSP, Functional Programming, Object-Oriented Programming, Delegation

1. INTRODUCTION
Open Sound Control (OSC) was originally designed as a message-passing format to facilitate exchange of control parameters between programs on different computers across a network. Since its release in 1997 [16] OSC has proven to be useful for message exchanges between processes on the same computer system and more recently within processing modules in the same program [17]. This paper shows how OSC messages can be used to provide composable, dynamic data types, to support generic, object-oriented and functional programming styles in dynamic, visual dataflow programming languages such as Max/MSP and PD.

2. Composable Aggregate Types
Max/MSP and PD are among the most popular programming languages for media computing and gesture signal processing for musical applications [6, 7]. An unfortunate legacy of the early success of these programs is their spartan support for data types and the lack of objects and a composable and extensible type system. These limitations are particularly problematic for the NIME community as projects increasingly involve complex gestural signal processing flows for large numbers of heterogeneous sensors and actuator types.

The solution advanced in this paper is to use Open Sound Control messages and native Max/MSP patches and externals to implement objects for dynamic, instance-based object-oriented programming (sometimes referred to as prototype-based programming). Self [15], ECMAScript [4], Javascript and NewtonScript [8, 12] are examples of languages using this programming style [1, 9, 11].

The ideas introduced here are embodied in a freely available collection of Max/MSP externals and patches known as the "o." library (pronounced "Oh dot"). We demonstrate applications of this library and new, productive programming techniques that leverage the high degree of composability [2] that emerges when delegation, aggregation and mapping techniques of object-oriented programming are melded to dataflow execution models.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

3. OSC Object Construction and Dispatch
3.1 Introduction
In prototype-based object-oriented programming objects are created from scratch (ex nihilo) or by cloning [14] and in some languages modifying an existing prototype object. The "o." library uses the cloning approach, allocating new memory, copying the contents of an inbound OSC message and modifying and adding to the copy as required (in the spirit of Kevo [9]). This approach follows the convention of Max primitive types, is easy to understand, avoids atomicity issues and allows programs to be easily distributed to multiple processors without the cost of managing references. It also invites a pure functional programming style with the well-known advantages of minimizing hidden state or stored values.

The conceptual steps from class-based object-oriented programming (OOP) to what we are doing with OSC here are small: concatenation of objects is sufficient for inheritance [9] and objects can serve as their own type definitions [5].

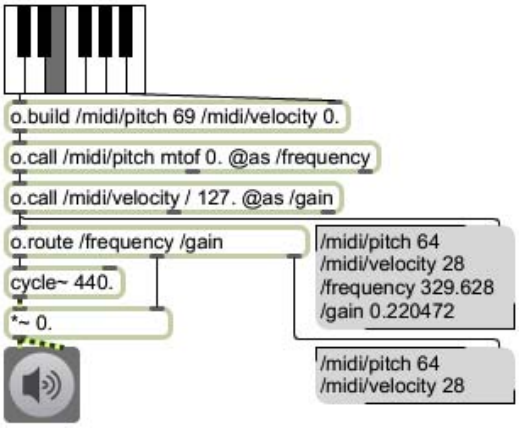


Figure 1. Aggregating values into OSC bundles.

3.2 Example
Figure 1 shows how o.build is used to create an interface to the Max keyboard object (kslider) that captures both legacy representations of the depression of a key on a musical keyboard as well as more contemporary ones. Bundling the

data from each outlet of the kslider better reflects the atomic binding of the two values implied by the gesture that created them than sending them as separate data at different times out of separate outlets. This common use of the o.build method is analogous to the "named associations" style of Ada function call arguments [13]. Instead of having to direct the right parameters at the right time to the appropriate inlet, parameters are named and bound together into a single bundle. This alternative to the positional association style of Max/MSP is also exploited in Jamoma [10].

The o.route method complements o.build to bring values out of the OSC bundle into the Max/MSP message world. Notice that the last outlet of o.route outputs a new bundle containing the unmatched elements of the original bundle. The "remainder" bundle can be further processed as the patch evolves–the essence of this delegation style of object-oriented inheritance. It is useful to contrast this approach with static class-based inheritance. The key difference is that in the delegation style new object types are created dynamically by simply adding new address/value pairs to existing objects. Programmers do not need to consult object definitions or API's to understand objects, their derivatives and promises: they simply look at the data in the objects themselves as they are formed and reformed using, for example, the gray UI Max object o.message which is analogous to the Max message box.

Just as Max wiring simulates physical wiring, OSC building and routing simulates scalable strategies used for wiring complex physical world systems, i.e. the labelling, color coding, bundling and bussing of wires.

4. Making OSC methods from Max patches

Using function-mapping approaches that are analogous to "map" and "apply" from Lisp, existing Max/MSP externals and patches can operate on data in OSC bundles. This eliminates the need for a large number of new speciality operators to be introduced and learned.

The most common scheme for this is implemented in the o.call method. This Max object instantiates a max patch internally according to its arguments and then routes named messages from incoming OSC packets to the internal max patch. Finally it gathers the output into an OSC bundle. Figures 1 and 2 illustrate this for various common scaling operations with floating-point division and the mtof (midi to frequency) function.

The o.call method uses prefix and suffix operators so it is syntactically closer to Lisp and other functional languages than to C. To clarify subsequent examples we note that the argument list comes first followed by the function description (a Max patch which o.call dynamically instantiates). Finally there may be closing attributes following an @ symbol. These are used to describe what to do with the results of the function call mapping. By default the result is bound to the same name as the first argument pattern. @as is followed by the names to be assigned to new elements that will be added to the incoming bundle. @prepending specifies a prefix to be added to the address of the first argument. These various conventions will be liberally used in the following examples.

5. Delegation-style Inheritance

In Figure 2 the example of Figure 1 has been augmented with the feature of pitch-dependent panning to illustrate how delegation can be used to add functionality to programs in a way that promotes reuse (the core benefit of object-oriented programming).

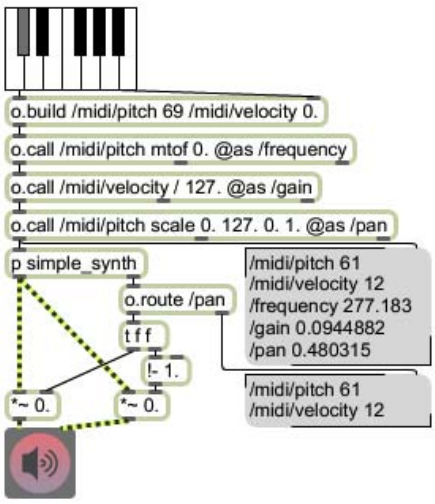


Figure 2. Delegation-style inheritance.

The objects in the top most grey box create a new bundle that includes the contents of the incoming bundle adding a new /pan value computed from the pitch in the incoming bundle.

The key to reuse is that neither the original patch assembling the description of the keyboard gesture nor the synthesizer need any changes to support the new /pan parameter. The additional sound functionality is added by using the delegation outlet (conventionally the rightmost) of the synthesizer patch. These new functionalities can of course be encapsulated as required. Note that the /pan route operation delegates its unmatched bundle inviting future inheritance.

6. External Sources/Sinks of OSC Bundles

External sources and sinks of OSC data include the venerable udpsend and udpreceive objects and slipOSC for serial-wrapped OSC (typically from USB serial devices).

The new o.io externals replace these Max functions by enumerating (o.io.discover) and wrapping (o.io) data from all the core I/O subsystems of OS/X computers as OSC messages. This sort of wrapping functionality is already partially addressed by programs such as Osculator (http:/osculator.net/) and Glovepie (http://glovepie.org). Unfortunately there is measurable and potentially troublesome variance in the delay of messages via these programs. The o.io object minimizes these by time-tagging the data using the lowest-level APIs to get as close as possible to the actual time the data was acquired. The o.io method already supports core popular protocols HID, UDP, TCP, MIDI, serial and proprietary API's such as the one provided for the built-in laptop motion sensors and multitouch trackpads.

The o.io method was carefully designed for extensibility so that new API's and device types can be easily added. Bundles from o.io typically contain the raw data from the API and then one or more overlays of higher-level interpreted data according to the device. For example some HID protocol devices provide entries in a table with useful names to substitute for the parameter numbers of the core data stream.

6.1 OSC Bundle Methods

Certain operations on bundles are clumsy to do by breaking them up into Max/MSP native types and reassembling them with o.build. These include merging, unions, intersections and accumulation for which the o.var method is provided. The o.if method is unusual in that it only inspects the contents of a bundle thereby avoiding a copy operation as it directs the bundle out of the "true" or "false" outlet according to the evaluation of a conditional expression.

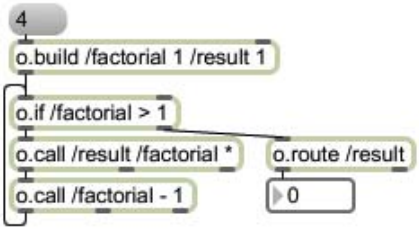


Figure 3. Recursion and o.if

The factorial calculation of Figure 3 illustrates o.if in action and how recursion is compactly done with "o." methods. Note that the evolving state of this computation is traceable by simply collecting the bundles recursively passed back. The concentration of observable state into bundles turns out to be a very productive programming technique, minimizing bugs that are hard to find because of state hidden within Max externals and patches.

7. Gesture Signal Processing with "o."

This section elaborates a complete gesture signal processing application by analysing the patch of figure 4 from top to bottom:

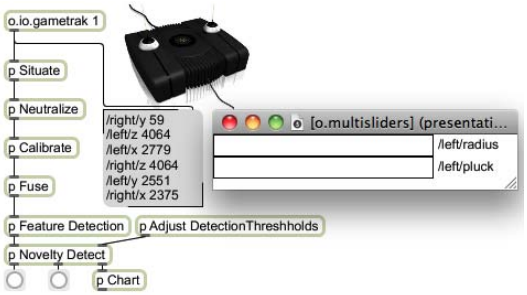


Figure 4: Feature Extraction

The source of analysed gesture data is a popular controller for experimental music called the Gametrak [3]. It provides data from the unwinding of retracting cords passing through the centers of a pair of joysticks. In the following sections we will trace the series abstractions encountered as packets move from top to bottom in this patch.

7.1 Situate

The HID encoding of values from the Gametrak is designed according to the viewpoint of the inventors of HID and their imagined uses for the Gametrak. We use the term "situate" to refer to the process of complementing this deferred agency of the hardware builders with the meaning the user of the Gametrak and Max patch can attribute according to their immediate situation. In the example shown this involves renaming.

```
o.rename /hid/Game-Trak-V1.3/1/15 /left/x
o.rename /hid/Game-Trak-V1.3/1/16 /left/y
o.rename /hid/Game-Trak-V1.3/1/17 /left/z
```

The appearance of x,y,z suggests the user's comfort with cartesian coordinate conventions. Another user might prefer the terms NS, WE, and Extension.

7.2 Neutralize

The value stream from this device (as with MIDI) confronts us with particular implementation choices: integers and the domain 0-4095. We neutralize this using the unit intervals [0-1] or [-1 1], the latter being useful in this case to represent directional deviations from the center of the joystick. These intervals are easy to scale by multiplication.

Regular expressions are used to match both the left and right addresses and to precisely call out a different range for x or y, or z.

```
o.call /*/z scale 4095 0 0. 1.
o.call /*/[xy] scale 0 4095 -1. 1.
```

This demonstrates the value of dynamic method routing and a surprising conciseness. The combination of wiring and patterns takes care of what is typically done more verbosely in lexical programming languages using terms such as lambda, self, this,or with.

7.3 Display

The named sliders displaying some of the values in the neutralized packet (in Figure 4) were built using o.multislider implemented using the same functional programming strategies of o.call while in addition tiling out the user interface.

7.4 Calibrate

Here we "taint" the domain of the neutralized gesture measurements by mapping them to a calibrated frame with extension in meters and positions as angles. We use the "prepending" attribute to add this interpreted value to the neutralized one rather than replace it.

```
o.call /*/[xy] * 30. @prepending /degrees
o.call /*/z * 2. @prepending /meters
```

This is an example of designing for reuse–a key aspect of composability. This calibration increases the potential for future reuse of the OSC packet (i.e. the object) without requiring knowledge of this future and without imposition of a complex interface.

7.5 Fuse

A simple sensor fusion is performed by combining the x,y axis data to create a radius and rotation. These are added to the copy of the incoming bundle.

```
o.call /left/x /left/y expr "sqrt($f1 * $f1 + $f2 * $f2)" @as /left/radius
o.call /left/x /left/y atan2 @as /left/orientation
```

7.6 Feature Detector

This feature detector computes the direction of a "stirring" gesture on the left string of the Gametrak. The algorithm is simply to look at the sign of the derivative of the rotation of the gesture. The first step is to use o.remember to build a packet containing the incoming packet and its predecessor. The elements of these two packets are distinct because the name "/was" is prepended to all the data in the old bundle. This avoids the complexity of the classical alternative: pointers or references. The difference operator is simply composed from Max/MSP's subtraction primitive:

```
o.remember unit delay
o.call /left/orientation /was/left/orientation - 0. @as /delta/left/orientation
```

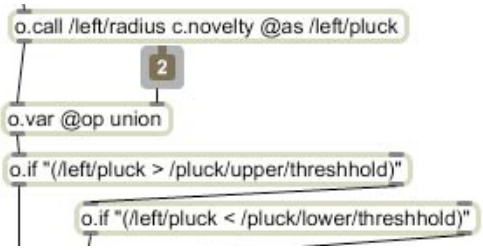
To complete the account here is the window function at the end of the feature detector:

```
p Difference
o.call /left/orientation > 0.001 @as /left/anticlockwise
o.call /left/orientation < -0.001 @as /left/clockwise
```

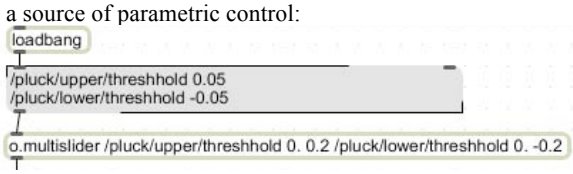
So far none of state of this computation is hidden. It is all available and traceable in the OSC bundles themselves. Although o.remember has to store a bundle internally, the stored contents are added to every outgoing bundle and flagged with the "/was" prefix. In the novelty detector of the next section we will stray slightly from this purely, functional approach but in a way that is still manageable.

7.7 Novelty Detector

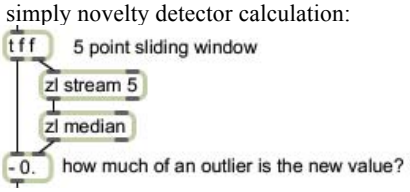
The basic algorithm is to clip the difference between the radial position of the left string and its median value inside a short sliding window.



As is typical of these simple detectors the threshholds of detection need to be adjusted using, for example, o.multisliders as a source of parametric control:



Here is the simply novelty detector calculation:



8. BlackBoxing

Hiding implementation details in modular, black boxes is a very effective technique but of little use unless the interfaces are documented. Our concatenating approach is interesting in that the name spaces can be designed so that the data itself emerging from the boxes describes the interface The packet emerging from the bottom of the example patch illustrates this.

```
/left/radius 1.194855
/was/right/y -0.967277
/was/left/radius 1.196872
/left/y -0.943346
/degrees/right/x -10.586081
/was/degrees/left/y -28.285715
/was/left/orientation 2.477973
/left/anticlockwise 1
/left/pluck -0.001591
/was/left/x 0.737241
/degrees/left/y -28.300365
/left/z 0.004640
/was/meters/left/z 0.008303
/left/orientation 2.480800
/delta/left/orientation 0.002828
/pluck/upper/threshhold 0.050000
/was/degrees/right/x -10.630037
/was/degrees/right/y -29.018314
/degrees/right/y -28.959707
/right/z 0.005372
/was/left/z 0.004151
/pluck/lower/threshhold -0.050000
/was/left/y -0.942857
/was/meters/right/z 0.009280
/was/degrees/left/x 22.117216
/right/x -0.352869
/was/right/z 0.004640
/right/y -0.965324
/degrees/left/x 22.000000
/left/clockwise 0
/meters/right/z 0.010745
/left/x 0.733333
/meters/left/z 0.009280
/was/right/x -0.354335
```

9. Conclusion

With the "o." library OSC messages are more than simply a new aggregate type for Max/MSP. They represent the glue necessary to integrate modern functional and object-oriented programming styles into a visual, dataflow language. Furthermore the time tags, atomicity and ordering semantics of OSC bundles promote productive development necessary for gesture signal processing and other reactive media programming applications.

10. Future Work

The "o." library has the foundational components to bring most modern programming paradigms to Max/MSP. Notable exceptions to this are reflectivity and parallelism. These both require significant changes in the Max kernel.

11. Dedication to Max Mathews

We dedicate this paper to the memory of Max Mathews who started us all out on computer languages for music and who mentored and inspired three generations of exciting work.

12. Acknowledgements

We gratefully acknowledge the support and encouragement of the McEnerney Endowment, Meyer Sound Laboratories, Pixar/Disney, the Concordia University Faculty of Fine Arts.

13. Bibliography

[1] Dony, C., Malenfant, J. and Bardou, D. Classifying Prototype-based Programming Languages. *Prototype-based Programming: Concepts, Languages and Applications*, 1998.

[2] Elliott, C. An embedded modeling language approach to interactive 3D and multimedia animation. *Software Engineering, IEEE Transactions on*, *25* (3). 291-308, 1999.

[3] Freed, A., McCutchen, D., Schmeder, A., Skriver Hansen, A.-M., Overholt, D., Burleson, W., Norgaard Jensen, C. and Mesker, A. Musical Applications and Design Techniques for the Gametrak Tethered Spatial Position Controller *SMC 2009*, 2009.

[4] Hansen, A.-M.S., Overholt, D., Burleson, W. and Jensen, C.N. Pendaphonics: a tangible pendulum-based sonic interaction experience *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, ACM, Cambridge, United Kingdom, 2009.

[5] Lieberman, H. Using prototypical objects to implement shared behavior in object-oriented systems. *ACM SIGPLAN Notices*, *21* (11). 214-223, 1986.

[6] Magnusson, T. and Hurtado, E. The Phenomenology of Musical Instruments: A Survey. *eContact!* , *10.4*, 2008.

[7] Magnusson, T. and Mendieta, E., The acoustic, the digital and the body: A survey on musical instruments. in, (2007), ACM, 94-99.

[8] McKeehan, J. and Rhodes, N. *Programming for the Newton: software development with NewtonScript*. Academic Press Professional, Inc. San Diego, CA, USA, 1995.

[9] Noble, J., Taivalsaari, A. and Moore, I. *Prototype-Based Programming: Concepts, Languages and Applications*. Springer, 1999.

[10] Place, T. and Lossius, T., Jamoma: A modular standard for structuring patches in Max. in *Proceedings of the 2006 International Computer Music Conference*, (2006).

[11] Smith, W. Class-based NewtonScript programming. *PIE Developers*, 1994.

[12] Smith, W. SELF and the Origins of NewtonScript. *PIE Developers magazine, July*, 1994.

[13] Taft, S. and Duff, R. *Ada 95 reference manual: language and standard libraries: international standard ISO/IEC 8652: 1995 (E)*. Springer Verlag, 1997.

[14] Taivalsaari, A. Delegation versus concatenation or cloning is inheritance too. *SIGPLAN OOPS Mess.*, *6* (3). 20-49, 1995.

[15] Ungar, D. and Smith, R., Self: The power of simplicity. in, (1987), ACM, 227-242.

[16] Wright, M. and Freed, A., Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. in *International Computer Music Conference*, (Thessaloniki, Hellas, 1997), International Computer Music Association, 101-104.

[17] Wright, M., Freed, A., Lee, A., Madden, T. and Momeni, A., Managing Complexity with Explicit Mapping of Gestures to Sound Control with OSC. in *International Computer Music Conference*, (Habana, Cuba, 2001), International Computer Music Association, 314-317.

SoundSaber - A Motion Capture Instrument

Kristian Nymoen, Ståle A. Skogstad
fourMs group - Music, Mind, Motion, Machines
Department of Informatics
University of Oslo, Norway
{krisny, savskogs}@ifi.uio.no

Alexander Refsum Jensenius
fourMs group - Music, Mind, Motion, Machines
Department of Musicology
University of Oslo, Norway
a.r.jensenius@imv.uio.no

ABSTRACT

The paper presents the SoundSaber - a musical instrument based on motion capture technology. We present technical details of the instrument and discuss the design development process. The SoundSaber may be used as an example of how high-fidelity motion capture equipment can be used for prototyping musical instruments, and we illustrate this with an example of a low-cost implementation of our motion capture instrument.

1. INTRODUCTION

We introduce the SoundSaber, a musical instrument based on optical infrared marker-based motion capture technology. *Motion capture* (mocap) involves recording motion, and translating it to the digital domain [10]. *Optical* motion capture means that the system is based on video cameras, and we distinguish between *marker-based* and *markerless* systems which work without markers. We will refer to musical instruments based on optical motion capture as *mocap instruments*.

Optical infrared marker-based mocap technology is superior to most other methods of motion capture with respect to temporal and spatial resolution. Some systems can track markers at a rate of more than 1000 frames per second, and in most cases they provide a spatial resolution in the sub-millimeter range. On the other hand, this technology is expensive, and better suited for laboratory use than for stage performances. A wide range of other less expensive and portable mocap technologies exists, like accelerometer-based sensor systems and computer vision. These provide different types of data, usually with lower frame rate and spatial resolution than optical infrared mocap.

A large amount of the research that is done in our lab involves the exploration of motion capture systems for musical interaction, ranging from high-end technologies to solutions like web-cameras and accelerometers. This involves studies of the different technologies separately, and also experiments on how the experience from interactive systems based on high-end mocap technology can be transferred to low-cost mocap technologies.

We present the SoundSaber as an example of how a seemingly simple sound synthesiser may become interesting through the use of high quality motion capture technology and an intuitive action-sound model. With a system that is able to register very subtle motion at a high sampling rate, it is possible to create an instrument that comes close to the control intimacy of acoustic instruments [11]. These ideas are presented through reflections that have been made while developing the instrument. Included in the presentation are some thoughts and experiences from how optical motion capture technology can be used to prototype new interfaces for musical expression.

In Section 2 we lay out a general theory on digital musical instruments and use of mocap for sound generation. Section 3 presents the SoundSaber, including considerations and evaluations that have been made in the process of development. In Section 4 we illustrate how the instrument was "ported" to another technology and compare the results to the original SoundSaber. Section 5 provides conclusions and directions for future work.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. MOCAP INSTRUMENT CONTROLLERS

Most digital musical instruments consist of a controller with sensors, a sound synthesiser, and a defined *mapping* between the control data from the sensors and the input parameters of the synthesiser [5]. Mocap instruments are slightly different in that the controller is separate from the sensor technology. This distinction between the sensors and the controller present an interesting opportunity because almost any object can be used to communicate with the mocap system: a rod, a hand, an acoustic instrument, etc.

This makes it possible to try out objects with different physical properties and shapes, hence also different *affordances*. In design literature, the affordance of an object is a term used to describe the perceived properties of how this object could possibly be used [6]. For an object used in a mocap instrument, the affordance may refer to a "pool" of different control actions that could be associated with it, e.g. whether it should be held with one or both hands. Following this, physical properties of the object, such as size, inertia, etc., will also influence how it can be handled. The possibility of quickly swapping objects may be a useful tool for prototyping new digital musical instruments.

The data from the motion capture system can be processed in several ways, see [1] and [10] for discussion on how motion capture data can be mapped to musical parameters. The GrainStick installation at IRCAM used mocap technology to generate sound in yet another way, using the metaphor of a virtual rainstick being held between two objects [4]. Our choices for data processing in the SoundSaber will be presented in Sections 3.2 to 3.5.

3. THE SOUNDSABER

The different components of the SoundSaber are illustrated in Figure 1. The position of the controller is captured by the motion capture system, which sends position data to a

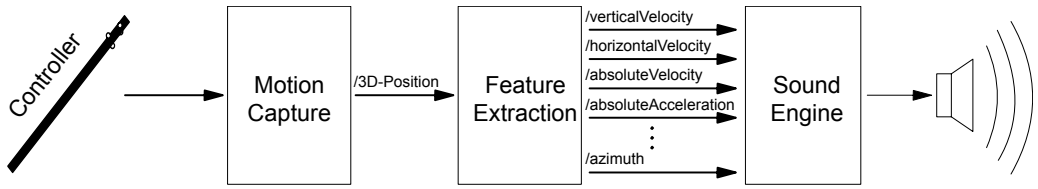


Figure 1: The figure shows the different parts of the SoundSaber instrument from controller, via motion capture technology and feature extraction to a sound synthesiser.

Max/MSP patch that calculates various features from the data. The features calculated by this patch are mapped to various control parameters in a synthesiser.

One of the advantages of digital musical instruments is that it is simple to try out different technologies for each part of the instrument. In our own work, we have experimented with different motion capture systems and controllers. Currently, we have two versions of the SoundSaber: The original one, based on optical motion capture, and another wii-controller (wiimote) implementation.

We will start the presentation of the SoundSaber by describing the controller, followed by a presentation of the motion capture technology, feature extraction and the synthesiser. Even though the different parts of the instrument are presented separately, they have been developed together, both simultaneously and iteratively.[1]

3.1 The controller

The SoundSaber controller that we are currently using is a rod, roughly 120 cm in length with a diameter of 4 cm, and is shown in Figure 2. Four markers are placed in one end of the rod, and the motion capture system recognizes these as a single *rigid object*, tracking position and orientation of the tip of the rod. The rod is heavy enough to give it a reasonable amount of inertia, and at the same time light enough so that it does not feel too heavy, at least not when it is held with both hands. The shape and mass of the rod also make it natural to perform large and smooth actions. We have observed that the majority of people who have tried the instrument performed gestures that imitate fencing. The reason for this may be their association of these gestures with the name of the instrument in combination with the physical properties and affordance of the controller.

Figure 2: The SoundSaber controller

3.2 Motion capture

We have been using different motion capture systems for the SoundSaber. Initially we used an 8-camera OptiTrack system from NaturalPoint, which can stream real-time data at a rate of 100 Hz. The OptiTrack software uses the proprietary NatNet protocol for data streaming. We used a client developed by Nuno Diniz at IPEM in Ghent for translating NatNet data to Open Sound Control (OSC) over UDP. OSC simplifies the communication between the motion capture system and the layers for feature extraction, mapping and sound synthesis.

More recently, we have been using a high-end motion capture system from Qualisys. This system has a higher spatial resolution than OptiTrack, and it is able to stream data at higher sampling rates. The Qualisys system also has native support for Open Sound Control.

[1]For video examples of the SoundSaber, please visit `http://www.youtube.com/fourmslab`

3.3 Feature extraction

We have implemented a tool in Max/MSP for real-time feature extraction from position data. Our approach is similar to the Motion Capture Music toolbox, developed by Dobrian et al. [1], with some differences. Our tool is structured as one single module, and outputs data as OSC messages. OSC formatting of these features simplifies the mapping between the motion features and the control features in the synthesiser.

Thus far, difference calculations, dimensionality reduction and transformations between different coordinate systems have been implemented. Based on a three-dimensional position stream the patch calculates:

- Velocity in a single direction, e.g. vertical velocity
- Velocity in a two-dimensional subspace, e.g. horizontal velocity
- Absolute velocity, as the vector magnitude of the three velocity components
- Change in absolute velocity
- Acceleration in a single direction
- Absolute acceleration
- Polar equivalent of the cartesian input coordinates, providing horizontal angle, elevation, and distance from the origin

3.4 Sound synthesis

As the name *SoundSaber* suggests, we initially had an idea of imitating the sound of the lightsaber from the Star Wars movies. The development of the synthesiser was more or less a process of trial and error to find a sound that would have some of the perceptual qualities that are found in the lightsaber sound.

The SoundSaber synthesiser is implemented in Max/MSP. Figure 3 shows a schematic illustration of the synthesiser, where a pulse train (a sequence of impulses or clicks) with a frequency of 1000 Hz is sent through two delay lines with feedback loops. The delay times for the delay lines can be adjusted by the user, resulting in variations in harmonic content. Furthermore, the output from the delay lines is sent to a ring modulator where it is modulated by a sinusoidal oscillator. The user can control the frequency of this oscillator in the range between 40 and 100 Hz. The ring modulated signal and the output from the delay lines are added together and sent through an amplitude control, then another feedback delay line and finally through a bandpass filter where the user controls bandwidth and frequency.

3.5 Mapping

Several considerations have been made regarding the action-sound relationship in the SoundSaber. Naturally, we have

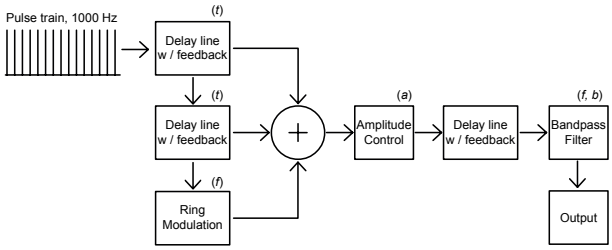


Figure 3: The SoundSaber synthesiser. Letters in parentheses denote user-controllable parameters: (t) = time, (f) = frequency, (a) = amplitude and (b) = bandwidth.

not been limited to mimicking action-sound relationships from traditional musical instruments, but at the same time we do appreciate some of the constraints that acoustic instruments provide. For instance, the sound of an instrument is almost always the result of an energy transfer from a sound-producing action to mechanical vibrations.

Since our approach to the physical design of this instrument has been simple, using only the position of a single point on the controller as the basis for feature extraction, we have chosen a simple approach when mapping motion features to control parameters. This is what Hunt and Wanderley call *explicit mapping*, meaning direct couplings between motion features and control parameters in the sound synthesiser [2].

When designing mapping for a mocap instrument, it is important to understand what the motion features actually describe. Motion features calculated from a stream of data describing position of a controller in a room can be one (or a combination) of the following:

Relative to the room meaning that the axes of the room influence the motion feature. An example of this is the vertical velocity component of the motion.

Relative to the controller itself typically referring to difference calculations, e.g. the absolute velocity.

Relative to another controller describing how the controller relates to other controllers in the room. For instance the distance to another SoundSaber.

In the current SoundSaber implementation, we have only used data that describes the controller in relation to the room or to itself. But we believe that the perspective of how the instrument relates to other controllers presents interesting possibilities in making collaborative musical instruments.

One of the considerations we have made is regarding motion in the horizontal plane. Should it make a difference whether the instrument is being moved along the X-axis or the Y-axis? In our opinion, the SoundSaber should respond equally whether the musician is on the left side or the right side of a stage, and also behave in the same manner no matter which direction the performer is facing, as is the case for any hand-held acoustic instrument. Therefore we reduced the two dimensions of horizontal velocity to a single absolute horizontal velocity, and let this mapping govern one of the timbral control parameters in the synthesiser (the delay time of the first delay line).

Vertical motion, on the other hand, is different. Our previous experiments have shown that people tend to relate vertical motion to changes in frequency, such as changes in pitch and spectral centroid [7, 8]. No matter which direction the performer is facing, gravity will act as a natural reference. In light of this, we have chosen to let the vertical position control the frequency of the ring modulation and the bandpass filter, and the vertical velocity control the delay time of the second delay line.

Another action-sound relationship which has been confirmed in our previous experiments, is the correspondence between velocity and loudness [7]. Hunt and Wanderley noted that increased input energy is required to increase sound energy in acoustic instruments, and received better results for a digital musical instrument where users had to feed the system with energy to generate sound, rather than just positioning a slider to adjust sound level [2]. With this in mind, we wanted an increase in kinetic energy to result in an increase in sound energy. Therefore, we let the absolute velocity control the amplitude of the synthesiser.

We have implemented a simple mapping for sound spatialisation. Spatial sound is obviously related to the room, so we used motion features related to the room in this mapping. More specifically, we sent the polar position coordinates of the rod to a VBAP control system [9], so the musician can control sound spatialisation by pointing the SoundSaber towards different loudspeakers.

3.6 SoundSaber evaluation

Neither the feature extraction, the explicit mapping strategy, nor the synthesiser of the SoundSaber are particularly sophisticated or novel by themselves. At the same time, after observing how people interact with the instrument, we feel confident to say that such interaction is engaging for the user. We believe that the most important reason for this are the considerations that were made to obtain a solid coupling between control actions and sound.

In addition to the rod, we tried using three other objects for controlling the SoundSaber synthesiser. For two of these, we simply changed the rod with another object and used the same motion capture technology, meaning that the only difference was the object itself. First, we tried a small rod, which was best suited for single-hand use, and also had less inertia and thus higher mobility. Second, we tried using a small handle with markers. This handle reduced the distinction between the controller and the performer, because the motion of the controller was basically equal to the hand motion of the performer. Both of these solutions were less satisfying than the large rod because the loudness control in the synthesiser had a fairly long response time, making it more suitable for controllers with more inertia. Also, the deep and full sound of the SoundSaber works better with a larger object. Third, as mentioned above, we made an implementation of the SoundSaber using a Nintendo Wii controller which will be discussed in more detail below.

Furthermore, we believe that the considerations of how motion features related to sound were important. The use of vertical position (which is only relative to the room) to adjust spectral centroid via a bandpass filter, and of absolute velocity (which is only relative to the object itself) to control loudness appeared to work well.

Nevertheless, the complexity of the control input, and the motion capture system's ability to capture motion nuances are perhaps the most important reasons why it is engaging to interact with the SoundSaber. Even though separate motion features were selected and mapped to different control parameters, the motion features themselves are related to each other. As an example, consider what happens when the performer makes a change in vertical position of the rod to adjust the spectral centroid. This action will also imply a change in the motion features "vertical velocity" and "absolute velocity".

When the spatial and temporal resolution of the motion

capture system is high, the instrument responds to even the smallest details of the performer's motion. For a reasonably sized object like the SoundSaber rod, we are satisfied with a spatial resolution of 1 mm and a frame rate of 100 Hz, but for smaller and more responsive objects we might require even higher resolution to capture the nuances of the actions these objects afford.

4. TOWARDS PORTABILITY

Because of the expensive hardware, the implementation of the SoundSaber based on optical motion capture is not available to everyone. One motivation for this research is to make instruments that are based on high-end technology available to a broader audience. Thus, we need less expensive and preferably also more portable solutions.

Of the many affordable sensor solutions, we chose to use a Nintendo wii-controller (wiimote) for our low-cost implementation. The wiimote provides a different set of control possibilities than optical motion capture, and the major challenges with porting the SoundSaber to the wiimote are related to processing the data from the controller and mapping strategies. A survey by Kiefer et al. ([3]) showed that the wiimote could be well suited for continuous control, which makes it an interesting test case for the SoundSaber.

4.1 Wiimote implementation

We used OSCulator[2] for communication between the wiimote and the computer. OSCulator provides estimates of orientation and absolute acceleration of the wiimote.

Orientation data can be seen as similar to the position data from the motion capture system, in the sense that it describes a state of the device within a single time-frame. Because of this similarity, change in orientation was mapped to the amplitude control. Although ideally the orientation data from the wiimote should not change unless there was an actual change in the orientation of the wiimote, the fact is that these values changed quite a lot even for non-rotational motion. Because of a significant amount of noise in the data, we used one of the push-buttons on the wiimote as an on/off button, to prevent the instrument from producing sound when the controller was lying still.

The angle between the floor and an imagined line along the length axis of the wiimote is called *pitch*. We let this value and its derivative control the synthesis parameters that originally were controlled by vertical position and vertical velocity, meaning the first delay line, frequency of the bandpass filter and the frequency of the ring modulator. Finally, we let the estimate of the dynamic acceleration control the second delay line in the synthesis patch.

4.2 Evaluation of the wiimote implementation

The wiimote implementation of the SoundSaber was, as expected, not as satisfying as the version based on optical motion capture. In our experience the orientation values needed some time to "settle". By this we mean that sudden actions affected these parameters quite a lot, and they did not settle at stable values until after the wiimote stopped moving. As a result, an action that was meant to cause a sudden increase in frequency would cause a sudden increase in loudness when the action started, and then a sudden increase in frequency when the wiimote was being held steady pointing up.

Using the tilt parameter *pitch* with the wiimote is conceptually quite different from the original mapping, where vertical position was used. However, we were surprised by how well this worked for slower motion. During a demonstration, one subject was moving the wiimote up and down with his arm fully stretched out, not realising that by doing this, he also pointed the wiimote up and down. The subject was puzzled by this and asked how we were able to extract vertical position values from the accelerometer in the wiimote.

In our opinion, the most important differences between the high-end implementation and the wiimote version are the size of the controller and the accuracy of the data. The wiimote data is too noisy for accurate control, and the size and shape of the wiimote afford one-handed, rapid impulsive actions, in contrast to the rod which is more suited for larger and slower actions. The wiimote implementation would probably benefit from using another synthesis module that is better suited for its affordances.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented the SoundSaber and our thoughts on how optical motion capture technology can be used for prototyping musical instruments. Our experience shows us that even a quite simple synthesiser and simple control signal are sufficient to create an interesting musical instrument, as long as the action-sound coupling is perceptually robust.

We will continue our work on the SoundSaber and other mocap instruments. It would be interesting to investigate whether the instrument would benefit from attaching an FSR. Furthermore, we see intriguing challenges and research questions related to developing the SoundSaber into a collaborative instrument, as well as an adaptive instrument that will adjust to different performers and situations.

6. REFERENCES

[1] C. Dobrian and F. Bevilacqua. Gestural control of music: using the vicon 8 motion capture system. In *Proceedings of NIME 2003*, pages 161–163, Montreal, Canada, 2003.

[2] A. Hunt and M. M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7(2):97–108, 2002.

[3] C. Kiefer, N. Collins, and G. Fitzpatrick. Evaluating the wiimote as a musical controller. In *Proceedings of ICMC 2008*, Belfast, Northern Ireland, 2008.

[4] G. Leslie et al. Grainstick: A collaborative, interactive sound installation. In *Proceedings of ICMC 2010*, New York, USA, 2010.

[5] E. R. Miranda and M. Wanderley. *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. A-R Editions, Inc., 2006.

[6] D. A. Norman. *The Design of Everyday Things*. Basic Books, New York, 1988.

[7] K. Nymoen. Comparing sound tracings performed to sounds with different sound envelopes. In *Proceedings of FRSM - CMMR 2011*, pages 225 – 229, Bhubaneswar, India, 2011.

[8] K. Nymoen, K. Glette, S. A. Skogstad, J. Torresen, and A. R. Jensenius. Searching for cross-individual relationships between sound and movement features using an SVM classifier. In *Proceedings of NIME 2010*, pages 259 – 262, Sydney, Australia, 2010.

[9] V. Pulkki. Generic panning tools for max/msp. In *Proceedings of ICMC 2000*, pages 304–307, 2000.

[10] S. A. Skogstad, A. R. Jensenius, and K. Nymoen. Using IR optical marker based motion capture for exploring musical interaction. In *Proceedings of NIME 2010*, pages 407–410, Sydney, Australia, 2010.

[11] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26:11–22, September 2002.

[2] `http://www.osculator.net/`

A modulation matrix for complex parameter sets

Øyvind Brandtsegg
Music Technology
NTNU, Department of Music
NO-7491 Trondheim
oyvind.brandtsegg@ntnu.no

Sigurd Saue
Music Technology
NTNU, Department of Music
NO-7491 Trondheim
sigurd.saue@ntnu.no

Thom Johansen
Q2S Centre of Excellence
NO-7491 Trondheim
thomj@alumni.ntnu.no

ABSTRACT
The article describes a flexible mapping technique realized as a many-to-many dynamic mapping matrix. Digital sound generation is typically controlled by a large number of parameters and efficient and flexible mapping is necessary to provide expressive control over the instrument. The proposed modulation matrix technique may be seen as a generic and self-modifying mapping mechanism integrated in a dynamic interpolation scheme. It is implemented efficiently by taking advantage of its inherent sparse matrix structure. The modulation matrix is used within the Hadron Particle Synthesizer, a complex granular module with 200 synthesis parameters and a simplified performance control structure with 4 expression parameters.

Keywords
Mapping, granular synthesis, modulation, live performance

1. INTRODUCTION
Digital musical instruments allow us to completely separate the performance interface from the sound generator. The connection between the two is what we refer to as *mapping*. Several researchers have pointed out that the expressiveness of digital musical instruments really depends upon the mapping used, and that creativity and playability are greatly influenced by a mapping that motivates exploration of the instrument (see e.g. [1]). In fact, experiments presented by Hunt et al [10] indicate that "complex mappings can provide quantifiable performance benefits and improved interface expressiveness". Rather than simple one-to-one parameter mappings between controller and sound generator, complex many-to-many mappings seem to promote a more holistic approach to the instrument: "less thinking, more playing".

The importance of efficient mapping strategies becomes obvious when controlling sound generators with a large number of input parameters in a live performance context. An interesting strategy proposed by Momeni and Wessel employs geometric models to characterize and control musical material [12], inspired by research on multidimensional perceptual scaling of timbre [9][16]. They argue that high-dimensional sound representations can be efficiently controlled by low-dimensional geometric models that fit well with standard controllers such as joysticks and tablets. Typically a small number of desirable sounds are represented as high-dimensional parameter vectors (e.g. the parameter set of a synthesis algorithm) and associated with specific coordinates in two-dimensional gesture space. When navigating through gesture space new sounds are generated as a result of an interpolation between the original parameter sets weighted by their relative distance in the space. Spatial positions can easily be stored and recalled as presets, and the gesture trajectories lend themselves naturally to automation.

We have adopted their strategy in a complex digital musical instrument designed for live performance with granular synthesis [2]. This particular synthesis engine [6] offers a wide range of time-based granular synthesis techniques found in Curtis Roads book *Microsound* [15]. The synthesis model requires some 40 parameters, but in addition we add modulation and effects for a total of over 200 parameters. The parameter vector not only represents a specific sound, but also contains information on how manual expression controllers and internal modulators may influence the sound. As a result navigation in gesture space actually modifies the mapping and hence changes the instrument itself.

This concept relates to Momeni's discussion of modal and non-modal mappings [13]. The former refers to mappings where the same gesture produces a variety of different results depending on instrument mode. Modal mappings provide richer control, but introduce state-dependent actions that may confuse the performer. In our case the modes are not discrete configurations, but rather a continuum of possible instrument states controlled and interpolated from the same gesture space as the sound itself.

The key mechanism for integrating this rich behavior into the mapping is the *dynamic modulation matrix*. The general idea is to regard all control parameters as modulators of the sound generator, and to do all mapping in one single matrix. The modulation matrix defines the interrelations between modulation sources and synthesis parameters in a very flexible fashion, and also allows modulation feedback. The entire matrix is dynamically changed through interpolation when the performer navigates gesture space.

In this paper we develop the concept of modulation matrix with simple examples. We then describe a particular implementation of the matrix within the audio programming language CSound. Finally we present the Hadron Particle Synthesizer as an example of a live performance instrument combining geometric interpolation and the modulation matrix.

2. THE MODULATION MATRIX
The origin of the modulation matrix is found in the patch bays of old analogue synthesizers, where patch cords interconnected the various synthesizer modules. In 1969 the English company Electronic Music Studios (EMS) [8] introduced the VCS3 with a unique matrix patch system to replace the clutter of patch wires (see front left panel in Figure 1 under). Signal routing was accomplished by placing small pins into the appropriate slots in the matrix.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

Figure 1. The EMS VCS3 synthesizer [8]

The concept is still common in software synthesizers and plug-ins. A relevant example is the Matrix Modular 3 synthesizer from Native Instruments [14] that offers a granular synthesis module, an integrated sequencer, various modulators and a modulation matrix that links them all together (Figure 2). The matrix is configurable, but there is no mechanism to dynamically interpolate from one matrix configuration to another, which is a cornerstone in what we propose.

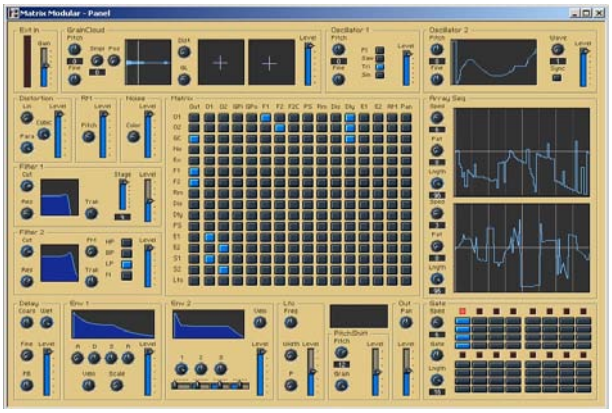


Figure 2. Native Instruments Matrix Modular 3 [14]

Software-based matrix modulation typically includes a list of modulation *sources*, a list of modulation *destinations*, and "slots" for each possible connection between source and destination. As an extension of the straightforward routing in classic patch bays software matrices often provide two controls between modulation source and destination:

- *Scaling coefficient*: The amount of modulation source reaching the destination.
- *Initial value*: An initial, fixed modulation value.

3. THE *MODMATRIX* OPCODE

Our implementation of the modulation matrix is available as an *opcode*[1] in the audio processing language CSound, with the name *modmatrix* [5]. The opcode computes a table of output values (destinations) as a function of initial values, modulation variables and scaling coefficients. The i'th output value is computed as:

$$out_i = in_i + \sum_k (g_{ki} * m_k)$$

where out_i is the output value, in_i is the corresponding initial value, m_k is the k'th modulation variable and g_{ki} is the scaling coefficient relating the k'th modulation variable to the i'th output.

In the following three sections we will provide some basic examples of *modmatrix* configurations. The modulator variables are assumed to be in the range 0.0 to 1.0 for unipolar signals (e.g. the signal from a user interface slider, called *manual expression control* here), and -1.0 to 1.0 for bipolar signals (e.g. the signal from an LFO). Amplitudes in the examples are assumed to be in range 0.0 to 1.0, and frequency values are given in Hz.

3.1 Simple modulator mapping

As a simple example of a modulation matrix we will use 2 parameters and 2 modulators. Each parameter has an initial value (the parameter value before modulation is applied), and the influence of each modulator signal to a parameter is computed by adding the modulator value to the parameter value. A scaling coefficient is applied to each modulator signal at each matrix mixing point (see Figure 3). With the specific coefficients used here the LFO will add a value of 0.4 to the oscillator amplitude when the LFO is at its peak value (if the LFO is bipolar, 0.4 will be subtracted when the LFO is at its minimum value). Oscillator frequency will also be affected by the LFO, with a maximum offset of 40Hz from the original oscillator frequency. The manual expression control affects oscillator amplitude (with a maximum offset of 0.6), and to a small degree also affects oscillator frequency (max offset of 5 Hz).

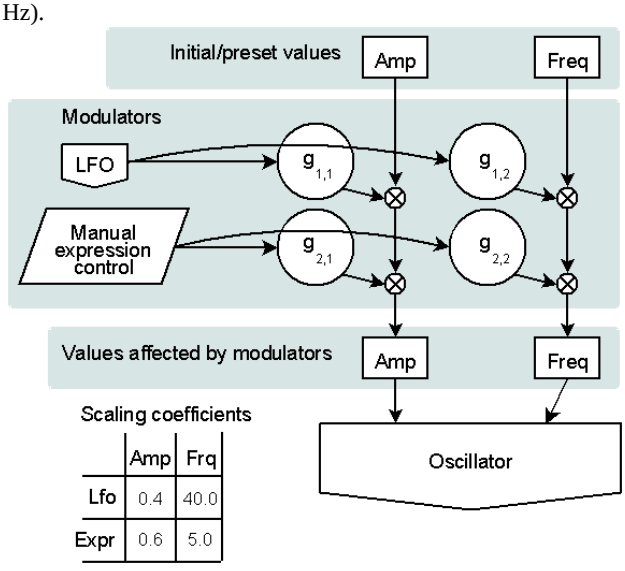


Figure 3. Parameters (Amp/Freq) for a simple oscillator modulated by one LFO and one manual expression control.

3.2 Modulator mapping with feedback

In this example, the parameter set has been extended to include amplitude and frequency for the LFO, and we enable modulator feedback in the matrix (see Figure 4). The modulator mappings to oscillator amplitude and frequency are the same as in the previous example. The LFO output affects its own frequency (by a maximum deviation of 7 Hz), and the manual expression control affects the LFO amplitude by a maximum offset of 0.4. Modulator feedback is known to be used in systems such as the

[1] An opcode is a basic CSound module that either generates or modifies signals.

Sytrus softsynth from Image Line [11]. Modulator feedback implies that some modulation sources may take the role as modulation destinations as well and we get self-modifying behavior controlled by the modulation matrix. As with any other kind of feedback, modulator feedback must be applied with caution.

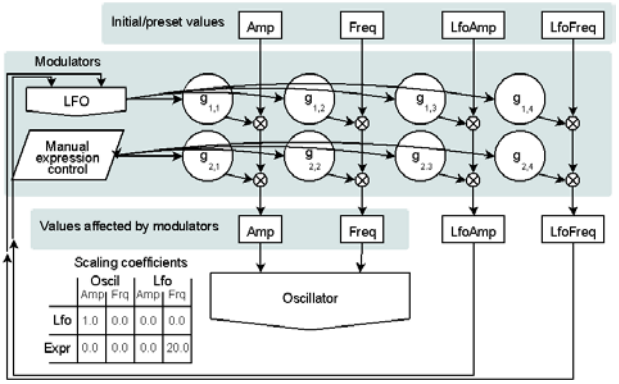


Figure 4. Modulation matrix with 4 parameters and 2 modulators. As some of the parameters are used in the synthesis of modulator signals, we have modulation feedback.

3.3 Dynamically modified mapping

As we operate with scaling coefficients stored in a table, we can dynamically alter the mapping in the modulation matrix by manipulating the values in the coefficient table. We can of course explicitly write values to the table, but more interesting: we can interpolate between different mapping tables.

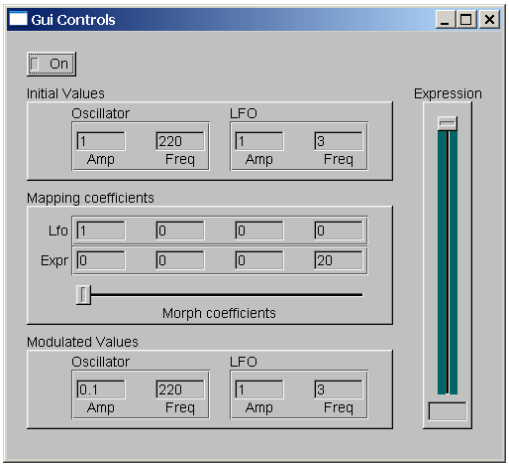


Figure 5. User interface for a simple synthesizer with a morphable modulation matrix [4].

In the following example [4] we will create a simple synthesizer with a modulation matrix as shown in Figure 4. We will enable interpolation between different tables of modmatrix coefficients, thereby morphing the mapping of the modulators. Our synthesizer GUI will have two user control sliders, one *expression* slider and one *morphing* slider. When moving the morphing slider, the modulation mapping will be dynamically changed. This changes how the LFO affects the sound, and it also changes the effect of the expression slider. To simplify our example all parameter values (oscillator amp and frequency, LFO amp and frequency) are fixed. Both the initial and the modulated values of these four parameters are shown in the user interface display (see Figure 5). A numerical example is shown in Figure 6.

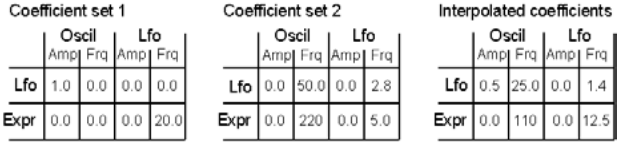


Figure 6. Interpolation between two tables of modulation matrix coefficients.

3.4 Implementation details

Due to the potentially large number of modulation sources for which modmatrix is designed to be used, optimization is a big concern. Fortunately, modulation matrices are typically sparsely populated, since only a small number of sources will be connected to any particular destination. Together with the fact that a given matrix is usually stationary between preset morphing states, this is an obvious way to improve performance.

There are several ways to deal with sparse matrices algorithmically, but most of these do not efficiently utilize the SIMD[2] capabilities present in the CPUs of all modern computers, so we have decided to apply a straightforward, but efficient method instead. Each time the synthesis environment knows that a preset interpolation, or any other activity altering the modulation matrix, is complete, it will signal modmatrix that the matrix is going into a temporarily constant state. The matrix will then be scanned for properties which can be eliminated, being for example entire rows and/or columns containing zeroes. A new modulation matrix will then be built lacking the redundant entries in question. From then on, the usual multiply and accumulate operations needed by a modulation matrix will be performed, skipping modulators and modulation targets which need not be taken into account. All modulation will be performed using this reduced matrix until the modulation matrix is again changed.

As long as the modulation matrix is undergoing change, the entire matrix is processed as is, still utilizing efficient SIMD processing. Looking into ways of efficiently leveraging the still sparse nature of the matrix in this morphing state is an area of future improvement, should the current method prove too inefficient.

Figure 7. Graphical user interface for the Hadron Particle Synthesizer.

4. THE HADRON PARTICLE SYNTHESIZER

As a more developed example of the modulation matrix, we will show how it's been utilized in the Hadron Particle Synthesizer[3]. Hadron is a complex granular synthesis device

[2] Single Instruction Multiple Data. Matrix computations will in most cases benefit measurably from use of these facilities.

[3] Hadron is freely available as a Max for Live device (march - 11) and as a VST plugin (fall 2011). It can be downloaded from www.partikkelaudio.com

with approximately 200 synthesis parameters. The underlying synthesis engine was built with a primary focus on flexibility of sound processing. The high level of flexibility also led to high complexity in configuration and control of the device. A simplified control structure was developed to allow real-time performance with precise control over the large parameter set using just a few user interface controls (see). The modulation matrix is essential to link simplicity of control to the complexity of the parameter set in this instrument.

4.1 Hadron internals

The basic parameters of granular synthesis can be considered to be *grain rate*, *grain pitch* and *grain shape*, as well as the audio *waveform* inside each grain. Hadron allows mixing of 4 source waveforms inside each grain, with independent pitch and phase for each source. The source waveforms can be recorded sounds or live audio input. The grain rate and pitch can be varied at audio rate to allow for frequency modulation effects, and displacement of individual grains allows smooth transitions between synchronous and asynchronous granular techniques. To enable separate processing of individual grains, a grain masking system is incorporated, enabling "per grain" specification of output routing, amplitude, pitch glissandi and more. A set of internal audio effects (ring modulators, filters, delays) allow further processing of individual grains. The Hadron Particle Synthesizer also utilizes a set of modulators for automation of parameter values. The modulators are well known signal generators, e.g. low frequency oscillators, envelope and random generators. In addition, audio analysis data for the source waveforms are used as modulator signals. Within Hadron, any signal that can affect a parameter value is considered a modulator, so signals from midi note input and the 4 manual expression controls (Figure 7) also counts as modulators. There is also a set of programmable modulator transform functions to allow waveshaping, division, multiplication and modulo operations on modulator signals.

The full parameter set for Hadron currently counts 209 parameters and 51 modulators. The granular processing requires "only" about 40 of these parameters, and a similar amount of parameters are used for effects control. The largest chunk of parameters is actually the modulator controls (e.g. LFO amplitude, LFO frequency, Envelope attack etc.) with approximately 100 parameters. All parameters and modulators are treated in one single modulation matrix with size 209 x 51.

4.2 Hadron control

Hadron makes use of a preset interpolation system, in many ways similar to techniques explored by Momeni and Wessel [12], but with some modification. We use static positioning of the presets. A preset is placed in each corner of a 2D "joystick" control surface. Another difference is that a preset not only contains parameter values, but also modulator mapping coefficients. This means that the effect of e.g. an LFO can change gradually from one preset to another. Similarly, the manual expression controls will have different effects in different presets. For example, if *Expression 1* controls *grain pitch* in one preset, it may control *grain rate* in another. Moreover, since the modulation matrix allows flexible one-to-many mappings and also nonlinear mapping curves via the modulator transform functions; both the routing and the scaling of a modulator may change between presets. The presets are manually designed to meet specific needs using a custom design tool, but the parameter space could possibly be explored using techniques similar to those suggested by Dahlstedt [7].

5. CONCLUSION

The article describes a flexible mapping technique realized as a many-to-many dynamic mapping matrix. A generalization of all control signals to be used as modulators allows for full flexibility of routing and mapping. Computationally efficient implementation of the modulation matrix allows practical use of large parameter and modulator sets. Dynamic mapping is achieved by interpolating mapping coefficients in the modulation matrix. Dynamic mapping can be combined with geometric models for parameter vector interpolation to create an instrument with simple controls, complex mapping and modal behavior. The complex mapping is not a goal in itself, but rather a result of the complexity of the parameter vector one wishes to achieve detailed control over. This complexity may lead to a higher learning threshold for the digital instrument, since the expression controls do not have fixed labels (like pitch bend, filter cutoff etc.). The lack of cognitive labeling of the instrument controls can be confusing for an unskilled performer, but as is the case with all musical instruments, practice is needed to achieve familiarity and skill. In fact, the lack of cognitive labeling may force a more intuitive approach to the instrument with an enhanced focus on listening. Our experiments on performance [3] using this mapping technique in the Hadron Particle Synthesizer shows that it can be used as a means of effective and intuitive instrumental control.

6. REFERENCES

[1] Arfib, D., Couturier, J., Kessous, L. and Verfaille, V. Strategies of mapping between model parameters using perceptual spaces. *Organised Sound 7*, 2 (2002): 127-144

[2] Brandtsegg, Ø. and Saue, S. Particle synthesis, a unified model for granular synthesis. Accepted paper at Linux Audio Conference 2011

[3] Brandtsegg, Ø and Waadeland, C. H. Studio sessions, duo improvisation with percussion (CHW) and *Hadron* (ØB): http://soundcloud.com/brandtsegg/sets/little-soldier-joe

[4] Brandtsegg, Ø. Example available as a Csound csd file at http://oeyvind.teks.no/ftp/modmatrix-example/modmatrix-simple-example.csd

[5] CSound opcode *modmatrix*. See documentation at: http://www.csounds.com/manual/html/modmatrix.html

[6] CSound opcode *partikkel*. See documentation at: http://www.csounds.com/manual/html/partikkel.html

[7] Dahlstedt, P. Dynamic Mapping Strategies for Expressive Synthesis Performance and Improvisation, in *Proceedings of the Computer Music Modeling and Retrieval (CMMR) Conference*, Copenhagen 2008

[8] Electronic Music Studios (EMS). Homepage (no longer updated) at: http://www.ems-synthi.demon.co.uk/

[9] Grey, J.M. Multidimensional perceptual scaling of timbre. *Journal of Acoustical Society of America 61*, 5(1977): 1270-1277

[10] Hunt, A., Wanderley, M. and Paradis, M. The importance of parameter mapping in electronic instrument design. *Journal of New Music Research 32*, 4(2003): 429-440

[11] Image Line. Homepage at: http://www.image-line.com

[12] Momeni, A. and Wessel, D. Characterizing and controlling musical material intuitively with geometric models. In *Proceedings of the New Interfaces for Musical Expression Conference (NIME-03)* (Montreal, Canada, May 22-24, 2003). Available at http://www.nime.org/2003/onlineproceedings/home.html

[13] Momeni, A. *Composing instruments: Inventing and performing with generative computer-based instruments.* Ph.D. thesis, University of California, Berkeley, CA, 2005

[14] Native Instruments. Homepage at: http://www.native-instruments.com

[15] Roads, C. *Microsound*. MIT Press, Cambridge, MA, 2001

[16] Wessel, D. Timbre space as a musical control structure. *Computer Music Journal 3*, 2(1979): 45-52

Sound Low Fun

Yu-Chung Tseng
eamusic.tseng@msa.hinet.net

Che-Wei Liu
drummerwei@gmail.com

Tzu-Heng Chi
davidchi0623@gmail.com

Hui-Yu Wang
huiyu18@gmail.com

National Chiao Tung University Master Program of sound and music Innovative Technologies
EE526 1001 University Road, Hsinchu, Taiwan 300, ROC

ABSTRACT

Sound Low Fun, a large sphere, is an interactive sound installation. The installation could produce low-frequency sound (Low Sound) to make people feel relax, to have "Fun"effect ("Fun" also pronounced close to Chinese word "放", which also means relax. This is our main concern and fundamental idea of the project. Our work present a sense of technology, and then we follow the structure by "C60" to divide into 32 blocks; Regarding the part of internal circuit design, we employed the force sensor and ADXL335 three-axis accelerometer connect with Arduino I/O and The Mux (Multiplexer) Shield, then it can produce different music with different lighting effects through Max/MSP programming. As music was concerned, we make use a type of meditative long-sustained low-frequency sound, accompanied by some transparency high-frequency sounds as sphere was shaked. When user presses, hugs and pushes the sphere, it trigger the soft low sound and lighting effects generated, as a result, user relieve his/her pressure eventually.

Keywords

Large-scale, interactive installation, low-frequency sounds, stress relief, Max/MSP computer music programming, Arduino

1. DESIGN CONCEPT AND CREATION IDEA

Our Installation to "Relieve Stress" for the design concept. We design by "Low Frequency" sound, and Interactive installation of large sphere with touch. Sound Low Fun can produce different music and light by user who push different blocks, and then try to make people relieve their stress. "Music" is one of the ways with curative effect of stress relief. Whatever Eastern culture or Western culture, they both have considerable research of "Music Therapy", The Eastern culture have five notes of traditional Chinese music "宮(Do) 商(Re) 角(Mi) 徵(Sol) 羽(La)" are same medical principles with five internal organs (liver, Heart, lung and Kidney). [1] And then, The Western culture also proved that music therapy created faster effects than traditional treatment for four times to eight times.

[1] Wikipedia-C60 http://zh.wikipedia.org/zh-tw/C60

2. IMPLEMENTATION METHODS AND TECHNIQUES

2.1. Interactive Design

This installation, "Sound Low Fun", has exhibition mode and interaction mode. In exhibition mode, has exhibition mode and interaction mode. In exhibition mode, the sphere can randomly generate scales and lights flashing, so that the installation can be regarding as a static device to present to all viewers. In Interaction mode, it has interaction between users and the installation, so that users can press or push the sphere to make it not only rotate or sway, but also generate or change music.

2.2. Appearance Design

This installation is a sphere. In order to distribute it completely, we adopt the structure of "Carbon 60"(Figure 4.) **[3]**, and then we divide the sphere into thirty-two blocks. Through pressing each block which has DIY sponge pressure sensor, it can send different arguments to trigger music. Additionally, we attach a piranha LED lights to sixty apices and use silver fabric to packing the sphere. When users press the sphere, the intensity of press can affect the brightness of the lights, and then making it light or shade.

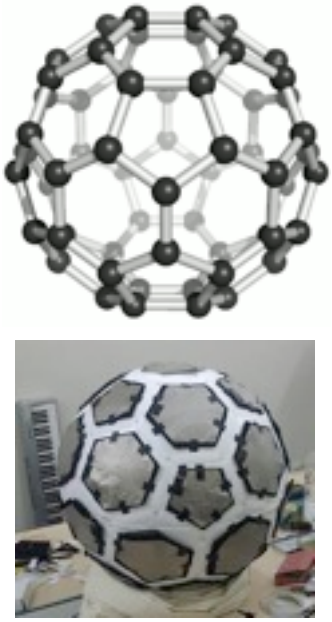


Figure 4. Structural drawing of Carbon 60[1]

2.3. Hardware Design

The DIY pressure sensor (Figure 7.) , which structure as follows : Using the conductive fabric as positive and negative, and placing conductive sponge among the conductive fabric as the resistance. The sponge deformation caused by people after they press it , the density become larger , the resistance become smaller so that the current that through the sponge will become larger, make the LED brighter. Briefly , people are able to control the LED of light shading by pressing the DIY pressure sensor.

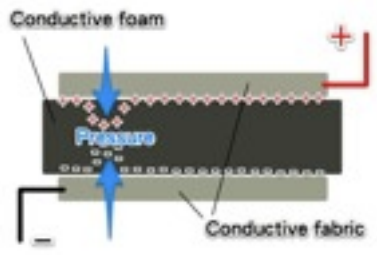


Figure 7. DIY Pressure Sensor

2.4. Software Design

In software, using Max/MSP communication between SimpleMessageSystem and Arduino. SimpleMessageSystem is a library which Arduino can send or receive the character and integer, and then analyse in Max/MSP.

3. MUSIC DESIGN

In music design, we use structure of “C60” to make the surface of sphere which have twelve pentagon and twenty Hexagon, each represents a different tone and harmony. We design our twelve-tone scale by “Pythagorean”, and then begin at F (Fà Cà Gà Dà Aà Eà Bà F#à C#à G#à D#à A#). Each notes are follow by order, and then backward to take three notes become harmony (FCG, CGD, GDA …). Followed later by analogy, and the order that we define “H”. (Figure 12.)

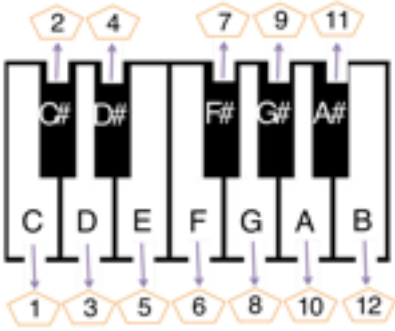


Figure 12. All the twelve notes of twelve pentagon can be a scale

There have one hexagon between every three pentagon (note), then the sound of hexagon is the harmony which combine with three pentagon (Figure 13.).

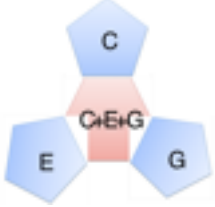


Figure 13. The hexagons of harmony constituted by three pentagons of single tone

We can choose one of the pentagons and regard it as pitch ”F”. Then basing on "order h ", corresponding to the two adjacent pentagons are pitch “C” and “G”. According to this rule, we can give each of ten pentagons a pitch, but the remaining two pentagons can not be Corresponded. It is the unexpected characteristic of the installation. Finally, twenty-six sides will be defined as different combinations of the harmony. (Figure 14.)

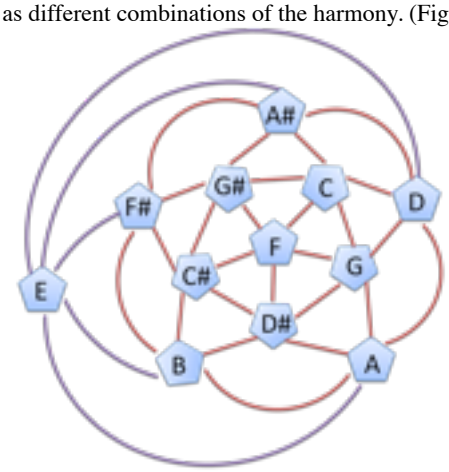


Figure 14. Pentagon corresponding to the pitch

Two low-frequency ambient sounds , one is pressing this spherical installation block, the other is with the LED flashes, a total of four. The sound source is used Ableton Live to the performance of the three sound sources, and the range of low-frequency range selected in the MIDI pitch on the note name C1 to B1. Through the communication by Ableton Live and Max/MSP, the midi signal sent to the Ableton Live to trigger the sound on and off. Further by interaction of touch, pressure and swing the spherical installation , do more effect in Ableton Live , make the sound more rich and colorful.

4. CONCLUSIONS

"Sound Low Fun" is a new digital art installation which integrating audition,visual and tactile.Through touching or pressing the surface of "Sound Low Fun", users can change the intensity of the lights and trigger low-frequency sounds.In this way, users do not only get the feedback of visual and audition, but also release the press.

5. REFERENCES

[1] Five notes of traditional Chinese music (Do)(Re)(Mi)(Sol)(La)” are same medical principles with five internal organs (liver, Heart, lung and Kidney)
http://www.dfg.cn/big5/yspd/jzxsh/47-wy-gong-1.html
[2] The music therapy are faster than traditional treatment for four times to eight times.
http://tinyurl.com/3e8gw98
[3] Structure of "Carbon 60"
http://zh.wikipedia.org/zh-tw/%E7%A2%B360
[4] The 2th K.T. Creativity Award Silver
http://140.115.78.29/rctedcontest/wp/?page_id=56
[5] RedBall
http://redballproject.com/taipei/1174/moca
[6] Music of relieve stress
http://www.windmusic.com.tw/shop/edm/edm0812_relaxation/001.htm

Autonomous New Media Artefacts (AutoNMA)

Edgar Berdahl
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
Stanford, CA, USA
eberdahl@ccrma.stanford.edu

Chris Chafe
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
Stanford, CA, USA
cc@ccrma.stanford.edu

ABSTRACT

The purpose of this brief paper is to revisit the question of longevity in present experimental practice and coin the term *autonomous new media artefacts* (AutoNMA), which are complete and independent of external computer systems, so they can be operable for a longer period of time and can be demonstrated at a moment's notice. We argue that platforms for prototyping should promote the creation of AutoNMA to make extant the devices which will be a part of the future history of new media.

Keywords

autonomous, standalone, Satellite CCRMA, Arduino

1. INTRODUCTION

For many decades, artists and engineers have been designing custom electronic interfaces for new media. Recently, this field has become especially popular as evidenced by the growth of the New Instruments for Musical Expression (NIME) conference, SIGGRAPH, other conferences, special sessions, and even attendance at massive do-it-yourself exhibitions such as the Maker Faire. Perhaps part of this trend is due to the wealth of documentation and tips now available on the Internet as well as the relative ease with which such interfaces can be prototyped. In this paper, we emphasize the value of creating prototypes that we call *autonomous new media artefacts*, or AutoNMA.[1] Because they are complete and independent of external computer systems, they can be demonstrated at a moment's notice, and they can be operable for a longer period of time.

Architects of some major edifices in the nineteenth century sometimes thought of the "future ruins" which they were creating. For example, an artist's rendition of the Bank of England was commissioned which depicted its future history [3]. Our interest is similarly related to processes of time, in this case the decay and erosion of our interfaces in new media. Most of the following discussion will pertain more specifically to sound, which is the specialty of our laboratory; however, we believe that the implications are the same for all new media (visual, haptic, etc.) that require significant computation.

[1]We would like to thank Wendy Ju for discussions on the subject as well as suggesting the term *autonomous*.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. MICROCONTROLLER WITH GENERAL PURPOSE COMPUTER

The most popular present platform for prototyping novel interfaces for sound creation consists of custom sensor circuits connected to a microcontroller, such as the Arduino, and a general purpose computer that receives sensor data over a data link and synthesizes sound [1]. Thus, the microcontroller's elegant interconnection with circuits is combined with the general purpose computer's ability to be programmed to perform many different kinds of sound synthesis. However, the general purpose computer does not operate autonomously. Consequently, over even only a few years, additional work is often required to keep prototypes running.

One the one hand, if a general purpose computer is *not* dedicated to the prototype, the prototype software may be affected by changes to other software on the computer, which may even be upgraded automatically or become infected with a computer virus, etc., potentially causing incompatibility with the prototype. Finally, the computer might become dedicated to a different prototype. No matter what, the general purpose computer will eventually require new hardware components or even become irreparable. Thus, eventually when demonstrating the prototype years down the road, additional effort will be required to preserve the data link between the microcontroller and the general purpose computer, especially as the data link technology may change.

The failure of custom new media devices is especially problematic for novel musical instruments, which may not survive long enough to allow a musician to develop virtuosic performance techniques, let alone repertoire. Some valiant designers continue to nurse their projects along over time to keep them alive despite changes to data link protocols and sound synthesis development platforms. However, the force of time is simply very strong. For this reason, we suggest that prototyping platform developers ensure that prototypes can be autonomous as a first step toward longevity.

3. ALTERNATIVE PLATFORMS

In this section, we present some alternative platforms for prototyping new media incorporating sound, but we do not attempt to survey all of the possibilities.

3.1 Smart Phones and MP3 Players

Smart phones and MP3 players are convenient because they are compact, battery powered, and incorporate significant computational power. Furthermore, some of them integrate accelerometers, touchscreens, etc., and a large number of applications have been developed to leverage these [7, 2]. For example, Figure 1 shows a smart phone mounted on a pair of headphones. The sensors in the phone are locked

Figure 1: Smart phone attached to headphones

to the motions of the head, so they can be employed for binaural synthesis and other augmented reality (AR) applications.

However, there is a value/lifespan problem with this platform. Most members of the new media community will prefer not to designate a new smart phone or MP3 player to live permanently within a prototype if it is needed for other applications. On the other hand, once a smart phone or MP3 player is older and of lesser value, its remaining lifespan will likely be quite short. Besides inevitable battery problems and possible hardware failure, smart phones and even many MP3 players are not autonomous—rather, they are designed to be connected to a service network, implying similar software change stability issues to those described in Section 2.

3.2 Satellite CCRMA

In contrast, Satellite CCRMA is another compact platform based on the power of smart phone processors, but is autonomous due to elimination of the battery and software updates. In addition, it is less expensive than the fanciest smart phones/MP3 players due to removal of unessential hardware. Furthermore, due to the presence of a standard USB bus and Linux support, it can be simply expanded and reconfigured by interfacing with Arduino, external sound interfaces, webcams, as well as (e.g. pico) video projectors. A picture of the platform is shown in Figure 2, where an Arduino Nano is stacked atop the Beagle Board-xM, which runs Satellite CCRMA via Ubuntu Linux at 1GHz. It supports floating-point operations natively and incorporates on-board Ethernet and stereo audio input/output codecs. For creating acoustic output, Satellite CCRMA can be connected to the Diamond MSP100B 4 Watts 2.0 Mini Rockers (see black speaker pair in Figure 2, left).

3.3 Small, Inexpensive, & Easy-To-Program: Microcontroller Alone

It is an interesting exercise to consider what platform is the smallest and least expensive while still being relatively easy to program. One possible example based on the Teensy microcontroller is shown in Figure 3, which we modified to operate off of a 3.3V power source by soldering the MCP1825 voltage regulator onto the underside.[2] It can be programmed over a USB connection using the Teensyduino add-on to the Arduino software environment.

The Teensy synthesizes sound by alternately turning an output pin on and off, which is connected to a speaker that consumes most of the energy from the 3.3V clock battery power source. By modulating the period of oscillation and pulse width, the Teensy can provide a large spectrum of fundamental frequencies and some variation in the timbre. Better control over the sound could be obtained by using a pulse width modulation pin with a low-pass filter, or better yet a commercial digital-to-analog converter (DAC) chip.

[2] `http://www.pjrc.com/store/mcp1825.html`

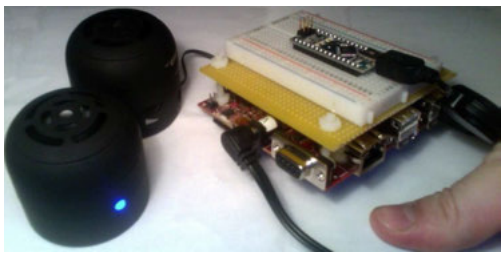


Figure 2: Satellite CCRMA

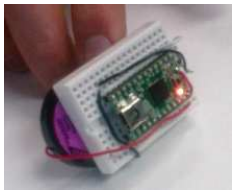


Figure 3: Teensy on breadboard (left) with 3.3V battery and loudspeaker on underside (right)

4. CAN NEW MUSICAL INSTRUMENTS GROW OLD?

In order to create a history of new media populated with actual artefacts that remain physically operable and easily demonstrable, we encourage the NIME community to take steps that allow their *new* musical instruments to grow *old*. For this reason, we believe that prototyping platforms should at least promote the creation of autonomous devices, ultimately ones which are long lived. In other words, we seek to create new media *artefacts*.[3] Perhaps we are far from the ideal, but it is interesting to consider what would be required to enable the archeology of new media. In our mind, the motivation is quite similar to that of electroacoustic music composers who wish to archive software, digital scores, paper scores, recordings, and descriptions of hardware setups [5, 6]; however, until now, there appears to have been little discussion of preservation of the hardware itself.

5. REFERENCES

[1] M. Banzi. *Getting Started with Arduino*. Make Books, Sebastopol, CA, 2008.

[2] N. Bryan, J. Herrera, J. Oh, and G. Wang. Momu: A mobile music toolkit. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Sydney, Australia, 2010.

[3] H. Dorey. Visions of Ruin: Architectural fantasies and designs for garden follies. In *Commentary and notes on the exhibition*. Ed. Christopher Woodward, Sir John Soane's Museum, London, UK, 1999.

[4] B. Kane. L'objet Sonore Maintenant: Pierre Schaeffer, sound objects and the phenomenological reduction. *Organised Sound*, 12:15–24, 2007.

[5] A. Moore. History and archival: The pitfalls of storage. In *Proceedings of the Colloque international informatique musicale*. Cite de la musique, Paris, France, Nov., 2009.

[6] B. Pennycook. Who will turn the knobs when I die? *Organised Sound*, 13:199–208, 2008.

[7] M. Rohs, G. Essl, and M. Roth. Camus: Live music performance using camera phones and visual grid tracking. In *Proceedings of the 6th International Conference on New Instruments for Musical Expression*. Paris, France, June, 2006.

[3] Also recently there has been some ambiguity in the usage of the term "sound object" [4].

Creating Musical Expression using Kinect

Min-Joon Yoo
Yonsei University, South Korea
Eng Bld C533, Sinchondong
Seoul, South Korea
debussy@cs.yonsei.ac.kr

Jin-Wook Beak
Yonsei University, South Korea
Eng Bld C533, Sinchondong
Seoul, South Korea
alleykat@cs.yonsei.ac.kr

In-Kwon Lee
Yonsei University, South Korea
Eng Bld C533, Sinchondong
Seoul, South Korea
iklee@yonsei.ac.kr

ABSTRACT
Recently, Microsoft introduced a game interface called Kinect for the Xbox 360 video game platform. This interface enables users to control and interact with the game console without the need to touch a controller. It largely increases the users' degree of freedom to express their emotion. In this paper, we first describe the system we developed to use this interface for sound generation and controlling musical expression. The skeleton data are extracted from users' motions and the data are translated to pre-defined MIDI data. We then use the MIDI data to control several applications. To allow the translation between the data, we implemented a simple Kinect-to-MIDI data convertor, which is introduced in this paper. We describe two applications to make music with Kinect: we first generate sound with Max/MSP, and then control the adlib with our own adlib generating system by the body movements of the users.

Keywords
Kinect, gaming interface, sound generation, adlib generation

1. INTRODUCTION
Kinect (http://www.xbox.com/kinect) is a new game interface for Microsoft's Xbox 360 game console. This interface enables users to control the console with their natural motion. This freedom is achieved by analyzing the image and sound of the users that the camera and microphone of the Kinect capture. Since this 'controller-free' interface has the ability to extend the degree of freedom and expressiveness of the users, many researchers and developers have tried to apply the interface in such a way that it not only controls the game console, but also controls their own applications. Through their efforts, open source drivers for Kinect have recently been developed and released, and various applications of Kinect have now been presented.

In this paper, we introduce our system, designed to create and control music according to the user's body motion via Kinect. Through Kinect's ability to interpret a user's whole motion, users can control the sound generation system with the rich range of movement of the body.

First, we extract skeleton data from the user's body. We can obtain the position and velocity of each joint of the user's body from the skeleton data. The joint data is then converted to MIDI data, which was defined previously. To translate the data, we implemented a Kinect-to-MIDI data translator.

Since the data from the user's body is now translated to MIDI data, general programs responding to the MIDI commands can be used to create sound or visualization. We tested Max/MSP for this purpose. We created various sounds with parameters converted from body motions in Max/MSP. We then applied this interface to our adlib generation program. The movements of the entire body were interpreted to generate control parameters for adlib.

2. KINECT
Kinect, officially launched in October 2010, is the first to achieve the elaborate full-body control with 3D motion capture. This interface is based on technology developed by Rare and PrimeSense for game technology and image-based 3D reconstruction. Aided by these technologies, this interface provides several useful functions to enable natural interaction.

Kinect consists of three devices: the RGB camera, the depth sensor, and the multi-array microphone (Figure 1). The camera can output video at 30 Hz frames with 8-bit VGA resolution (640 x 480 pixels). The monochrome depth sensor can then sense the depth information, also with VGA resolution with 11-bit depth, which provides 2,048 levels of sensitivity. The practical range limit of the sensor is 1.2-3.5 m (3.9-11 ft), and the sensing range is adjustable. By interpreting the data obtained from the camera and depth sensor, we can realize the 3D motion capture of the users. Kinect also has a microphone array consisting of four microphone capsules. These devices operate with each channel processing 16-bit audio at a sampling rate of 16 kHz. You can find more detailed information about the devices on the Kinect homepage and Wikipedia (http://en.wikipedia.org/wiki/Kinect).

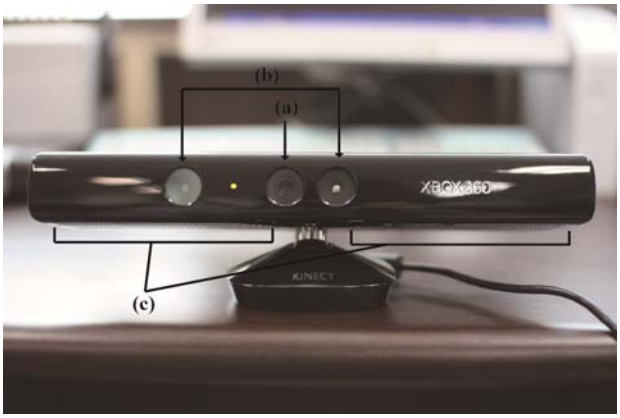


Figure 1. Kinect consists of three components: (a) RGB camera, (b) depth sensors, and (c) microphone array.

3. SYSTEM DESCRIPTION

We used modules developed in OpenNI (http://www.openni.org/) to connect Kinect with a PC. The PrimeSense's *NITE* provided useful APIs for the manipulation of naïve data. We also use FAAST (http://projects.ict.usc.edu/mxr/faast/) to easily access the joint data extracted from the body motion. The FAAST streams the user's skeleton data over a VRPN server (http://www.cs.unc.edu/Research/vrpn/).

We then implemented a Kinect-to-MIDI convertor (see Figure 2). This program listens to messages from the VRPN server embedded in FAAST, and responds if proper messages are received. We predefined a mapping between the joint data and the MIDI data. Thus, the input joint messages are translated to MIDI messages using this mapping. Finally the MIDI messages are sent to a MIDI-IN port.

Our translator is operated in a similar way to the Wii-to-MIDI translators such as GlovePIE or OSCulator. We were inspired by these programs for creating music using game controllers.

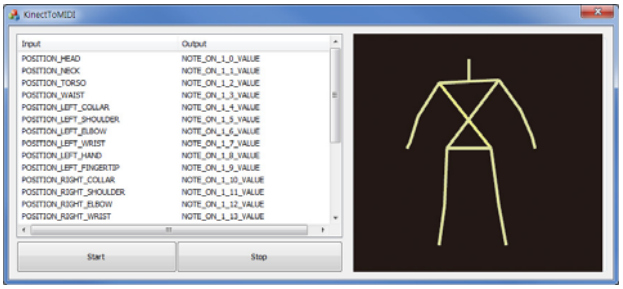


Figure 2. Kinect-to-MIDI convertor.

4. SKELETON DATA

FAAST streams a total of 24 skeleton joint data tracked in the input video. The joints are ordered corresponding to the OpenNI framework (Table 1). As you can see in this table, the skeleton data contains all the positions of the major parts of the body, and we can reconstruct the movement of the body approximately from the skeleton data.

Table 1. The skeleton joint order by OpenNI framework.

Sensor	Joint	Sensor	Joint
0	Head	12	Right Elbow
1	Neck	13	Right Wrist
2	Torso	14	Right Hand
3	Waist	15	Right Fingertip
4	Left Collar	16	Left Hip
5	Left Shoulder	17	Left Knee
6	Left Elbow	18	Left Ankle
7	Left Wrist	19	Left Foot
8	Left Hand	20	Right Hip
9	Left Fingertip	21	Right Knee
10	Right Collar	22	Right Ankle
11	Right Shoulder	23	Right Foot

5. APPLICATIONS

5.1 With Max/MSP

With our Kinect-MIDI convertor, any music application controlled MIDI data can be used to create music and sound. We chose the Max/MSP program to test the controllability of Kinect because of its efficiency in creating and modifying sound. We implemented a Max/MSP patch, which generates sounds by several parameters.

The skeleton data can be roughly divided into five segments: center (0-3), left arm (4-9), right arm (10-15), left leg (16-19), and right leg (20-23). Thus, we generated five sounds corresponding to the body segments, controlled by 4-6 parameters also corresponding to the joint data belonging to each segment. For example, one sound corresponding to the left leg had four parameters, and these parameters are controlled by the movements of the left hip (16), left knee (17), left ankle (18) and left foot (19).

We then mapped the velocity of the joints to the parameters of the sound. The velocity of a joint can be easily calculated by the difference between the position of the joint in one video frame and the position in the previous video frame. By using velocity data, users could change the sound more intuitively.

5.2 With Our Adlib Generator

We tested our system in our own adlib software. The software was originally implemented to generate an adlib sequence via the user's line drawing. The user drew lines using a mouse interface, and the adlib was generated by data obtained using the movement and position of the mouse. We extended the software by changing the input from mouse to Kinect.

The speed and pitch of the adlib was controlled by the average velocity and position of each joint, respectively. The scale of the adlib was then determined by the relative position of each end-joint (head (0), left fingertip (9). right fingertip (15), left foot (19), and right foot (23)). We predefined the positions of the end-joints generating a normal pose. Then, if the end-joints were closer to each other than the normal pose, (the user's pose shrunk), the adlib was created with a diminished scale. Also, the farther the end-joints were from each other, the tenser the scales used to generate adlib. Compared with the previous control using a mouse, this system can express more adlib styles because of the more intuitive controller. We are now conducting user surveys about the control method and resulting adlibs.

6. CONCLUSION

We first reported a system to generate and control sound with Kinect. Several open-source drivers and modules were used to extract the position data from the movement of a user's body. We then converted the data into the MIDI messages by using our Kinect-to-MIDI translator. Because the data from the body motion can be presented in MIDI messages, any music application responding to MIDI data can be used to create and control music. We first tested this method with Max/MSP software, then with our adlib generator. In our prototype tests, we could control the music more intuitively using body motions. We are exploring more applications and methods for controlling music using Kinect and also doing more theoretical research by studying literature related to analysis of movement. .

Making grains tangible: microtouch for microsound

Staas de Jong
LIACS, Leiden University
Niels Bohrweg 1, Leiden
staas@liacs.nl

ABSTRACT

This paper proposes a new research direction for the large family of instrumental musical interfaces where sound is generated using digital granular synthesis, and where interaction and control involve the (fine) operation of stiff, flat contact surfaces.

First, within a historical context, a general absence of, and clear need for, tangible output that is dynamically instantiated by the grain-generating process *itself* is identified. Second, to fill this gap, a concrete general approach is proposed based on the careful construction of non-vibratory and vibratory force pulses, in a one-to-one relationship with sonic grains.

An informal pilot psychophysics experiment initiating the approach was conducted, which took into account the two main cases for applying forces to the human skin: perpendicular, and lateral. Initial results indicate that the force pulse approach can enable perceivably multidimensional, tangible display of the ongoing grain-generating process. Moreover, it was found that this can be made to meaningfully happen (in real time) in the same timescale of basic sonic grain generation. This is not a trivial property, and provides an important and positive fundament for further developing this type of enhanced display. It also leads to the exciting prospect of making arbitrary sonic grains actual physical manipulanda.

Keywords
instrumental control, tangible display, tangible manipulation, granular sound synthesis

1. INTRODUCTION

1.1 Granular synthesis of musical sound, and its instrumental control

During the 19th and 20th centuries, newly developed technologies included increasingly practical methods to capture, transform and reproduce fragments of sound. When this is done for musical purposes, and the fragments involved have a brief duration of 0.1 s or less, the term *microsound* is often used [7]. In 1960, the composer Iannis Xenakis coined the term "grains of sound" in this context, also proposing a number of mathematically defined compositional tools for combining these grains into musical sound [10]. Currently, in granular synthesis a *grain* is defined as a sound fragment of duration 1 to 100 ms, resulting from a waveform signal shaped by an amplitude envelope. Over the years, composers increasingly have adopted granular techniques to create music, resulting in influential early works by Iannis Xenakis, Horacio Vaggione, Curtis Roads, Barry Truax, and others, and today granular sound synthesis is in widespread use.

Grain-based approaches to making musical sound were first implemented in a cumbersome process using analog magnetic tape technology. The subsequent revolution in the power and availability of digital computing technology, however, enabled the implementation and use of a series of increasingly sophisticated and powerful versions of granular sound synthesis [7]. It also enabled the introduction of implementations where the actions of instrumental control could occur simultaneously with the listening to their results [9]. Today, there are many such real-time implementations of granular sound synthesis available, often controlled using Graphical User Interfaces (GUIs) and the input from various types of MIDI controllers.

1.2 The interest of giving grains a dynamically instantiated tangible presence

One important use of the tangible aspects, in general, of instrumental control, in general, is display: to inform the human actions that are performed. Another important use is in defining *how* these actions can be performed, in the manipulations that are made possible. In the case of granular synthesis of musical sound, the object of such tangible display and manipulation will be the process of grain generation.

In existing systems, tangible display and manipulation are usually implemented using various types of general-purpose controller hardware, such as buttons, sliders, knobs, pads, and keys. These can then be used to initiate, modulate and terminate processes of grain generation in real time. However, the display and manipulation enabled by these components will not be very specific to the processes of grain generation that are controlled. Stages of tangible display and manipulation can be usefully set up to coincide with stages of grain generation. (E.g. as when overcoming the specific friction of moving a slider to a certain position, while this is being mapped to, say, the granular density.) However, this type of control is fundamentally limited, by the fact that it is not the process of grain generation itself that determines the tangible feedback.

In practice, these existing types of tangible display will give relatively little information about the grain generation in progress. As a consequence, for specific and detailed information, human operators will largely rely on the auditory display provided by the output of musical sound. This reliance has inherent disadvantages, e.g. in that the response of human actions to auditory feedback necessarily will be slower than the response to tangible feedback, making control less immediate [8].

For the above reasons, the existing real-time instrumental control of granular synthesis could be improved by using new forms of tangible display directly determined by the process of grain generation itself. These could provide the human operator with more information for her/his control, while this information could be made more specifically relevant; and

could be delivered more immediately than is currently the case.

One such new form of tangible display is represented by the set of interfaces described in [4]. In these, the sound resulting from manipulations with colliding, breaking, deforming and sliding items is used to trigger and parametrize the digital generation of sound grains in real time. The stated main motivation for this approach was to expand, for musical purposes, the sonic range of familiar tangible manipulations. However, the resulting forms of instrumental control also have the property that the grain-generating processes directly determine tangible display, giving the advantages above.

What remains to be done, however, is to give these advantages, in the same way, to the widely used algorithmic processes of grain generation running on digital computing hardware in general. Here, the generation of each granular sound fragment will happen according to a set of explicitly defined parameters. Usually, the values for these parameters are uniquely determined for each grain, at the moment of its instantiation, to then remain fixed for the rest of its duration. Therefore, in order for tangible display to provide information that is as complete as possible, it should be capable of providing each grain instance with its own, dynamically determined tangible representation.

Having identified some necessary and desirable characteristics for new forms of tangible display for algorithmic grain generation, this can now motivate and guide the investigation of concrete methods of tangible representation. Such investigation must also remain alert to possibilities for *manipulation*, since the possibilities that are identified for tangible display and tangible manipulation will together enable as well as delimit the designs than can ultimately realize improved instrumental control.

1.3 Approach: force output to the fingerpad

When implementing tangible display to dynamically represent separate grains generated by algorithmic processes, this will first require choices in anatomical location and means of delivery. The hands can be considered as the most versatile parts of the human body for sensing and manipulating the immediate tangible surroundings. For fine sensing and manipulation, the fingertips especially are used as the areas of contact, having the highest spatial resolution in the cutaneous (skin-based) sense of touch across the hand [5]. Such fingertip contact will often involve the fingerpad skin areas, which have been used for the instrumental control of musical sound over tens of millennia, e.g. to close the holes of flutes [1], pluck sounding strings, press keyboard keys, etc.

Here, we will consider flat, stiff surfaces, put in contact with the fingerpads to apply forces, controlled over time, to the fingers. In general, this can result not only in cutaneous but also in kinesthetic sensations of touch involving finger movement. We will use two general interfaces for touch in instrumental control of musical sound, which have been described elsewhere: the cyclotactor ("CT") [2] and the kinetic surface friction renderer ("KSFR") [3]. In the CT, the flat surface is attached to the fingerpad using a strap. Voluntary fingerpad movements are intended to happen only perpendicularly to the fingerpad's surface. In the KSFR, the flat surface is pressed down upon. Voluntary fingerpad movements are then limited to happen in parallel to the fingerpad's surface. This is shown in Figure 1, where the different types of intentional movement and applied force in the two interfaces are described and illustrated in more detail.

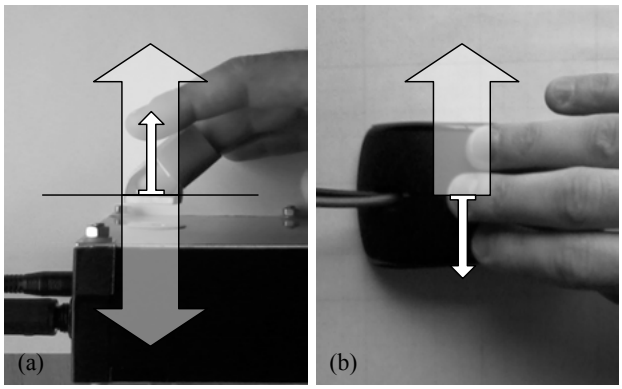


Figure 1. Fingerpad movements and applied forces in the two interfaces used. The large transparent arrows indicate directions of intentional movement; the small opaque arrows indicate the direction of the controlled force components that are applied.

(a) The CT setup: intentional movements are performed perpendicularly to the fingerpad surface. Regardless of whether the fingerpad is intentionally moved up, down or held still, the direction of applied forces will (here) be perpendicularly against the fingerpad.

(b) The KSFR setup: intentional movements are performed parallel to the fingerpad surface. Forces are applied during movement, and are opposed to the direction of movement (only one case is shown).

2. PILOT EXPERIMENT

2.1 Overview

To investigate possibilities for tangible display of granular synthesis using the two general methods of force delivery to the fingerpad, an informal pilot experiment was conducted. In it, both force magnitude and headphone sound output were controlled over time. Both were determined by the same variable, on/off master block impulse signal. This master block impulse controlled sound output by modulating a sine wave signal of a relatively high frequency, allowing the signal to retain pitch more easily for shorter impulse durations. The master block impulse controlled force output in a similar way, by modulating either a sine wave signal or a level maximum amplitude signal. Here, the sine wave used had a frequency of 250 Hz, placing it within the frequency region where the vibrational sensitivity of mechanoreceptors is highest [6]. To help create the impression of a single "grain event" occurring on both channels, millisecond latencies were adjusted so that the patterns in sound and force output would temporally coincide as much as possible. As cannot be seen in Figure 1, a single finger was used to contact the stiff surface of the KSFR during intentional movement. Tables 1 and 2 describe the experimental parameter values that were kept constant and those that were varied, respectively.

Table 1. Experimental parameters kept constant.

interval between successive grain event onsets:	1.00 s
sound block impulse maximum amplitude:	constant
sound carrier signal sine frequency:	4000 Hz
baseline force level:	0.14 N

Table 2. Experimental parameters that were varied.

interface:	CT / KSFR
master block impulse duration:	100 / 50 / 10 / 1 ms
force block impulse max. amplitude:	1.00 / 0.72 / 0.43 N
modulated force signal:	constant / 250 Hz sine

2.2 Results

In the CT interface, both with and without headphone output, the non-vibratory force impulses generated seemed clearly perceivable for all of the impulse durations tested. The differences in these impulse durations, as well as the differences in amplitude at given impulse durations also seemed clearly perceivable. Of the vibratory force impulses, only the durations above 1 ms were considered, since only these would fit at least one vibration wave cycle (of duration 4 ms). For these durations, both the differences in duration and in amplitude seemed clearly perceivable. The type of sensation seemed to change with duration: at 100 and 50 ms, impulses seemed to give an impression of vibration, while at 10 ms, this changed to a pulsed sensation that seemed less distinct when compared to a non-vibratory impulse of the same duration.

In the KSFR interface, force impulses of duration 1 ms could not be considered due to a technical issue: at this duration, a mechanical effect in the housing of the device resulted in the perception of forces in the fingerpad also when it was being held still. This made it ambiguous whether apparently weak forces applied in parallel to the fingerpad surface during movement were being felt separately of this, or not. At the remaining durations of 10 ms and higher, however, these forces seemed well distinguishable both for the non-vibratory and vibratory force impulses. Also, both the differences in duration and in amplitude at each duration seemed perceivable. Here too, the type of sensation seemed to change with duration for vibratory force impulses: at 50 and 100 ms these gave an impression of vibration, while at 10 ms this again changed to a pulsed sensation, not unlike that produced by a non-vibratory impulse of the same duration.

In both interfaces, force impulses with larger amplitudes and durations were clearly able to influence position and speed input. In the CT interface, this resulted in vertical displacements of the fingerpad; in the KSFR interface, it resulted in the slowing down of intentional movements. Figure 2 below shows an example recording of output by the CT interface, at the 'microtouch' end of the temporal scale.

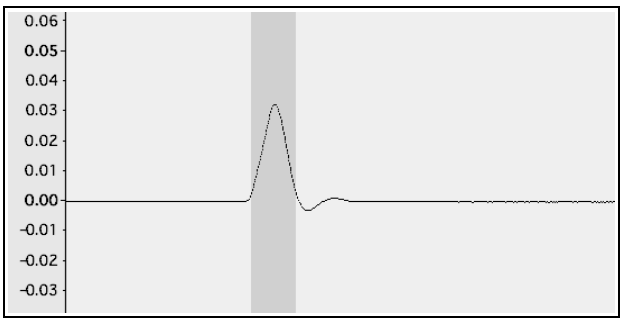


Figure 2. Microtouch output. Linear magnetic field strength recording, made during the application of a perceived force impulse by the CT interface. The grey vertical bar represents a duration of 1.0 ms.

3. DISCUSSION AND FUTURE WORK

The results of the informal pilot experiment indicate that the perpendicular and parallel methods of force delivery to the fingerpad as implemented in the CT and KSFR interfaces can be used for the tangible display of separate grain events, meaningfully operating in the same time scale that underlies the granular synthesis of musical sound. Specifically, it seems that variations in the amplitude and duration of applied force impulses could be used to dynamically mirror or display aspects of grain generation, starting from the level of separate grains. For longer impulse durations, it seems vibratory force could add an additional dimension to such display.

Of the two interfaces tested, it seems the CT is currently somewhat better positioned to display fine detail in applied forces developing over time.

It may seem self-evident that for successfully improved forms of instrumental control of granular synthesis to be realized, the musical sound output and tangible display of systems should be developed in close tandem. This seems the more so since signals to one sense can influence the perception of other signals to other senses in many ways: for example, vibrotactile stimulation influences the sensation of hearing a tone [11].

Can grains become manipulanda? One way towards this suggested by the results from the pilot experiment seems to be using the changes in displacement and velocity input caused by force impulse output – as net displacements will be the result of forces applied by both the interface and the user.

For this reason and the reasons stated at the beginning of this section, based on the fundamental motivating factors discussed in the introduction, it seems that the methods of dynamically applying force to the fingerpad presented here should be further investigated for their potential to enable new and appropriate forms of tangible display and manipulation for the instrumental control of granular musical sound.

4. REFERENCES

[1] Conard, N. J., Malina, M., and Münzel, S. C. New flutes document the earliest musical tradition in southwestern Germany. *Nature, 460* (Aug. 2009), 737-740.

[2] De Jong, S. Presenting the cyclotactor project. In *Proc. of the fourth international conference on tangible, embedded and embodied interaction (TEI'10)* (Cambridge, MA, USA, January 25-27, 2010). ACM, 319-320.

[3] De Jong, S. Kinetic surface friction rendering for interactive sonification: an initial exploration. In *Proc. of ISon 2010 - Interactive sonification workshop : Human interaction with auditory displays* (Stockholm, Sweden, April 7, 2010). KTH School of Computer Science and Communication, 105-108.

[4] Essl, G., and O'Modhrain, S. An enactive approach to the design of new tangible musical instruments. *Organised Sound, 11(3)* (2006), 285-296.

[5] Goldstein, E. B. *Sensation and Perception, 6th edition.* Brooks/Cole, Pacific Grove, CA, USA, 2002, 446.

[6] Marshall, M.T., and Wanderley, M. M. Vibrotactile feedback in digital musical instruments. In *Proc. of the 6th International Conference on New Interfaces for Musical Expression (NIME 06)* (Paris, France, June 4-8, 2006).

[7] Roads, C. *Microsound.* The MIT Press, Cambridge, MA, USA, 2004.

[8] Rovan, J., and Hayward, V. Typology of tactile sounds and their synthesis in gesture-driven computer music performance. In *Trends in Gestural Control of Music.* Editions IRCAM, Paris, France, 2000, 297-320.

[9] Truax, B. Real-time granular synthesis with the DMX-1000. In *Proc. of the 1986 International Computer Music Conference* 138-145.

[10] Xenakis, I. Elements of stochastic music. *Gravensaner Blätter, 18* (1960), 84-105.

[11] Yarrow, K., Haggard, P., and Rothwell, J. C. Vibrotactile – auditory interactions are post-perceptual. Perception, 37 (2008), 1114-1130.

Sound Selection by Gestures

Baptiste Caramiaux
IMTR Team
Ircam - CNRS
Paris, France
caramiau@ircam.fr

Frédéric Bevilacqua
IMTR Team
Ircam - CNRS
Paris, France
bevilacq@ircam.fr

Norbert Schnell
IMTR Team
Ircam - CNRS
Paris, France
schnell@ircam.fr

ABSTRACT

This paper presents a prototypical tool for sound selection driven by users' gestures. Sound selection by gestures is a particular case of "query by content" in multimedia databases. Gesture-to-Sound matching is based on computing the similarity between both gesture and sound parameters' temporal evolution. The tool presents three algorithms for matching gesture query to sound target. The system leads to several applications in sound design, virtual instrument design and interactive installation.

Keywords

Query by Gesture, Time Series Analysis, Sonic Interaction

1. INTRODUCTION

The study presented in this paper is part of a series of studies concerning the analysis of the relationships between movements and sounds for the design of virtual instruments and more generally for applications in sonic interaction. Consider the following scenario. A user imagines a sound that is too abstract to be described using words. Possibly, a skilled user should be able to sketch with the voice what the sound looks like. Here we consider the case where the person uses gestures. If the profiles drawn by the temporal evolution of the sound's characteristics is clear in the users' mind, they could try to gesturally "trace the sound" either in the air or on a surface. Thus, the goal of the proposed tool is to return a sound that is the most pertinent according to the tracing of the performed gesture. The problem is a particular case of "query by content" in multimedia databases. The input gesture is usually called the *query* and the resulting sound the *target*.

1.1 Background

The general problem of "query by content" in multimedia database was extensively studied and the literature is flourishing. In Music Information Retrieval (MIR), a particular case of "query by content" is the famous "query by humming" problem [4]. Query by humming system allows the user to find a song in a database by humming part of the tune. Most researches into query by humming use the notion of *contours* that is the the sequence of relative differences in pitch between successive notes. Another illustrative example is the "query by tapping" system [6] that allows the user to find a song by tapping the rhythm. This system is based on onset detection and temporal alignment.

On the gestural counterpart, there is a dramatic lack of literature about audio query by gesture systems in either the NIME or MIR communities. Previous works are more dealing with the inverting system that is analyzing which gesture is performed by a user while listening to a sound [5]. When trying to match a gesture and a sound, two problems occur: which features should we select for describing either the gesture or the sound? how can we fill the informational resolution gap between both signals?

1.2 Proposed system

Figure 1 illustrates the system for gesture-driven sound selection. A user performs a gesture that matches, at least from the user perspective, an abstract sound. After a preprocessing module, the system contains several algorithms for time series multimodal matching. Each algorithm retrieves a specific part of information in the relationship between gesture and sound. The algorithm is the choice of the user. The matching algorithm returns the sound index in the database together with a score that indicates the target pertinency. Finally the sound is played to the user.

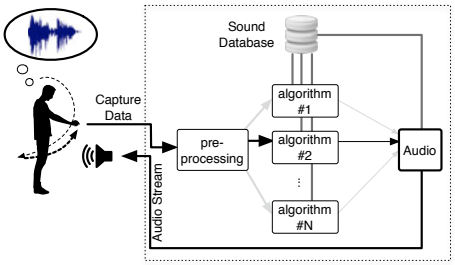


Figure 1: The proposed system. A sound that best matches the input gesture is found in a database. The matching depends on the algorithm used.

2. PROTOTYPE

In this section, we present the implementation. The algorithms used in the current version are reported in the next section. Then the available implementation in the Max/MSP software is described.

2.1 Algorithms

Each matching method allows for retrieving specific information in the relationship between gesture and sound.

Correlation-based selection

The method is based on the correlation between the input gesture parameters and the sound features [3]. The method

NIME'11, 30 May–1 June 2011, Oslo, Norway.

is similar to Principal Component Analysis but adapted for two datasets of different dimensionality. It is called Canonical Correlation Analysis (CCA). The algorithm finds the principal (or canonical) components that explain the most the covariance between the two datasets. Then, it returns two new sets of variables (for both gesture and sound) that are ordered from the most correlated (the first ones) to the less correlated (the last ones). Sound selection tool based on CCA allows for the selection of the predominant features (in terms of correlation) from both gesture and sound parameters. The first correlation coefficient (i.e the maximum) is used as the similarity score. A sound is selected if the variation of a combination of its features is similar to the variation of a combination of the gesture parameters. Since correlation is computed sample-by-sample, a high score also indicates that gesture is synchronous to the sound. Finally the sound is selected at the end of the gesture leading to the need to mark the beginning and the end (e.g. using a button).

Time-warping based selection

One can ask to preserve the inherent variability in gesture and choose as similarity criterion the global shape matching and the coherence between amplitudes. To that extent, the second strategy is based on temporal alignment of both multidimensional signals. The method returns a score that depends on whether the two signals are far from each other in terms of alignment and amplitudes. This method is an HMM-based technique that has been used for gesture recognition and following [2]. It is computationally efficient, multidimensional, real time and makes use of a simplified learning process. For sound selection tool, a sound is selected if the user performs a gesture that evolves similarly to the sound features but can be non-linearly time shifted. Here real time means that a sound is selected while the gesture is performing. However, it requires to previously select the features chosen to be matched.

An hybrid strategy

The algorithm is iterative and uses both correlation-based measure and temporal alignment. The strategy is to compute CCA between the user's gesture taken as input and all the sounds in the database. Then, we take the sound corresponding to the highest correlation coefficient and we apply a temporal alignment between the projected correlated variables. We then iterate using the aligned gesture and the original sounds in the database. The use of temporal alignment is two-fold. First it allows to better discriminate the candidate sound from the other. Second, it allows to precise which feature is actually predominant in the mapping user's gesture-to-selected sound. The iterative process is heuristic but results to always increase the correlation coefficient. Using this strategy allows for more temporal flexibility without constraining the system by fixing previously the features but is computationally time-consuming.

2.2 Implementation

The various algorithms are encapsulated in an application developed in the Max/MSP real-time programming environment (and MnM [1]). The sound pool uses MuBu [7] that contains N sounds together with their audio descriptors. These audio descriptors are directly computed in Max/MSP. The motion data are received by OSC allowing for the use of a wide range of interfaces. When the analysis is done, the program returns the index of the best matching sound belonging into the database, and it is visualized in the MuBu editor (see figure 2 for a screenshot of the tool).

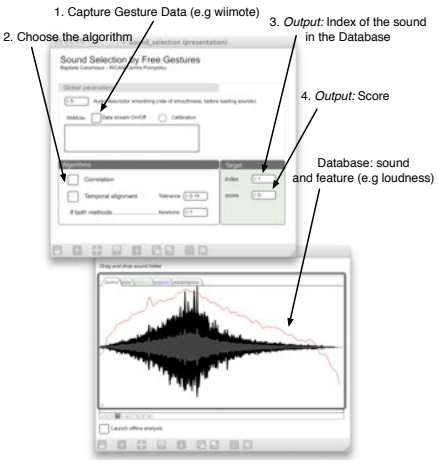


Figure 2: The Max/MSP patch for sound selection by free gestures. The example is given with one feature (loudness) per sound and is used with a WiiMote controller. The user can choose which algorithm is used for the time series matching.

3. CONCLUSION

In this paper, we presented an application allowing for sound selection driven by user's gestures. The application computes the similarity between the gesture and sound parameters' temporal evolution. The tool aims to embed several algorithms for time series matching. A version has been developed in the Max/MSP software and uses MnM.

Finally, we have recently investigated by an experimental study how people associate gestures to environmental sounds for which either the cause having produced the sound can be identified or not. This study will give important insights for the relationships between gesture and sound and will help for the design of new algorithms.

4. ACKNOWLEDGMENTS

We acknowledge partial support from the project Interlude -ANR -08-CORD-010 (French National Research Agency).

5. REFERENCES

[1] F. Bevilacqua, R. Müller, and N. Schnell. Mnm: a max/msp mapping toolbox. In *Proceedings of the 2005 conference on NIME*, pages 85–88. National University of Singapore, 2005.

[2] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. Continuous realtime gesture following and recognition. In *Lecture Notes in Computer Science (LNCS)*. Springer Verlag, 2009.

[3] B. Caramiaux, F. Bevilacqua, and N. Schnell. Towards a gesture-sound cross-modal analysis. *Lectures Notes in Computer Science, Springer-Verlag*, 2009.

[4] R. Dannenberg, W. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis. A comparative evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Society for Information Science and Technology*, 58(5):687–701, 2007.

[5] R. I. Godøy, E. Haga, and A. R. Jensenius. Exploring music-related gestures by sound-tracing: A preliminary study. In *Proceedings of the COST287-ConGAS 2nd International Symposium on Gesture Interfaces for Multimedia Systems (GIMS2006)*, 2006.

[6] J. Jang, H. Lee, and C. Yeh. Query by tapping: A new paradigm for content-based music retrieval from acoustic input. *Advances in Multimedia Information Processing*, pages 590–597, 2001.

[7] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, R. Borghesi, et al. Mubu & friends-assembling tools for content based real-time interactive audio processing in max/msp. In *Proceedings of the ICMC, Montreal*. Citeseer, 2009.

An Open Source Interface based on Biological Neural Networks for Interactive Music Performance

Hernán Kerlleñevich
LAPSo
Universidad Nacional de Quilmes,
Buenos Aires, Argentina
hk@lapso.org

Manuel C. Eguía
LAPSo
Universidad Nacional de Quilmes,
Buenos Aires, Argentina
me@lapso.org

Pablo E. Riera
LAPSo
Universidad Nacional de Quilmes,
Buenos Aires, Argentina
me@lapso.org

ABSTRACT

We propose and discuss an open source real-time interface that focuses in the vast potential for interactive sound art creation emerging from biological neural networks, as paradigmatic complex systems for musical exploration. In particular, we focus on networks that are responsible for the generation of rhythmic patterns.The interface relies upon the idea of relating metaphorically neural behaviors to electronic and acoustic instruments notes, by means of flexible mapping strategies. The user can intuitively design network configurations by dynamically creating neurons and configuring their inter-connectivity. The core of the system is based in events emerging from his network design, which functions in a similar way to what happens in real small neural networks. Having multiple signal and data inputs and outputs, as well as standard communications protocols such as MIDI, OSC and TCP/IP, it becomes and unique tool for composers and performers, suitable for different performance scenarios, like live electronics, sound installations and telematic concerts.

Keywords

rhythm generation, biological neural networks, complex patterns, musical interface, network performance

1. INTRODUCTION

The brain is the most complex organ in nature, and it stands among all living tissues by its time-organized action [1]. The collective behavior of millions of interconnected neurons conforms organized and stratified rhythm systems that interact with each other. These interactive rhythm layers can reveal network architectures and also produce patterns which are not understandable from the individual behavior of a each neuron, whose action can be explained by biophysical and biochemical processes. Even in its minimal expression, simple neural systems as invertebrate ganglia [5] or central pattern generators (CPGs) in the spinal chord [7] are complex systems that can exhibit nonlinear behavior, emergent properties, and the combination of regular activity and unpredictability in the long run.

In this work we employ simple, yet dynamically rich, neural systems to develop new interfaces for generative music composition and performance. Interfaces based on these properties can go far beyond the usual ones, since they are able to generate structured chains of events that can interact, not only with the performer, but also with each other. Even with little or no external input control, this kind of systems displays a robust variety of stable and unstable time structures. With flexible mapping strategies their use for musical expression has virtually no limit.

We developed a biological neural network interface that is able to generate patterns of diverse degree of complexity, and that can be extended to multiple coordinated centers, as in rhythm networks in the brain. The interface is named SANTIAGO, after the renowned Spanish physiologist Santiago Ramon y Cajal. It consists of a modular patch developed in PD, including a core for biological neural network simulations and diverse input/output modules that can be mapped to the desired musical parameters, as pitch, timbre, beat, etc.

There were many previous efforts employing other complex systems as new flexible interfaces, including Markov chains, cellular automata, L-systems, chaotic oscillators, generative grammar, and genetic algorithms, among others, and some of them are available nowadays as tools for music creation[1]. However, applications for music composition and performance based on the dynamics of biological neural networks[2] are less explored. A former effort that explores biological inspired networks for granular synthesis is reported in [6]. Our development though is focused on a different time scale, corresponding to rhythm and note generation.

In a previous work [4] we presented Santiago at an early stage of development and gave details about the biological models and their implementation. This article addresses what we consider as relevant interface and performance issues and shows several examples of simple networks giving rise to complex rhythmic outputs. The presentation is structured as follows: in section 2 we give a theoretical background for the interface, in section 3 we develop the interface design, section 4 is about possible performance scenarios, section 5 discusses our preliminary analysis of the interface and in section 6 we conclude with some remarks and ideas for further development.

2. BACKGROUND

Neural networks are composed by computational units (neurons) linked by synapses. Depending on how these units are modelled we can have more simplified and unrealistic artificial neural networks (where the complexity resides in

[1]For a list see e.g. `http://en.wikipedia.org/wiki/Generative_music`

[2]A brief distinction between this type and Artificial Neural Networks can be found in section 2 (Background)

NIME'11, 30 May–1 June 2011, Oslo, Norway.

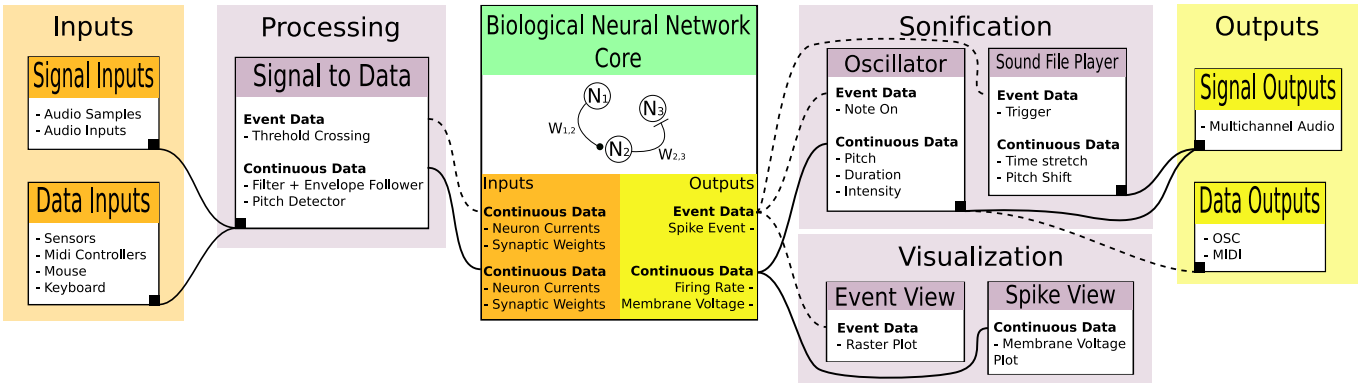


Figure 1: Santiago's data and signal flow across its modular structure. Signal and data inputs may be processed to extract desired features and scale their values whithin ranges chosen by the user. Then, the data streams or events are fed into the system core, consisting of a neuron panel with completely configurable units and a connectivity matrix, where the network is created and designed. The generated data can be mapped to sonification parameters of ad-hoc synths and sound-file players routed to a multichannel audio system (if available) as well as sent via MIDI and OSC outputs. Event and Spike visualization tools are also available, for a clear visual feedback of the network activity.

the connectivity pattern only), or biologically inspired networks, where the intrinsic dynamics of the neurons are taken into account. In this last case, the activity of the units is given by a sequence of electrical pulses (spikes), that can be modified via the synapses, either by excitatory or inhibitory action of other neurons. We take this second approach, using a variety of neuron models displaying different intrinsic behaviors and responses to stimulation (see fig. 3).

Much of the dynamical richness of the interface comes precisely from the choice of the mathematical neuron model for the network core. The model was proposed by E. Izhikevich [3], and is described by a system of two differential equations (Eqs. 1a and 1b) and one resetting rule (Eq 1c):

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I(t) + I_{syn} + \xi(t) \quad (1a)$$

$$\frac{du}{dt} = a(bv - u) \quad (1b)$$

$$if\, v \geq 30\, mV,\ then \begin{cases} v \longleftarrow c \\ u \longleftarrow u + d \end{cases} \quad (1c)$$

The voltage membrane of the neuron v and a recovery variable u are the dynamical variables. Four dimensionless parameters (a, b, c, d) and one input current $I(t)$ determine the type of behavior exhibited by the model. The neuron receives synaptic inputs from other neurons through I_{syn}. A stochastic component or noise current $\xi(t)$ can also be added. The spike mechanism works by resetting the variables v and u when the voltage reaches some fixed value. Despite its simplicity, this system can replicate the dynamical behavior of most neuron types. Subtle changes in the four parameters or the input current values of the neuron model can give rise to very different rhythmic behaviours.

Neuron units are connected unidirectionally to other neurons through synapses, that could be either excitatory or inhibitory. If two neurons are connected via an excitatory (inhibitory) synapse, the spike events in the signal-passing neuron are transformed to positive (negative) synaptic current pulses, with a characteristic exponential decay. These pulses are delayed and added to the total synaptic current of the target neuron (I_{syn}).

It is interesting to note, that with these units it is possible to build small networks that can give rise to complex patterns of spikes, even using very few neurons (as few as two, see figure 4). These patterns (that can be observed in real brains via multi-electrode recordings) include neural beats, synchronization, and periodic or almost recurrent behavior in different time scales. A particular case of this last case are rhythmic networks, that are often encountered in the animal realm as pacemakers and central pattern generators (CPG). Activities such as walking, running, jumping, swimming, breathing and chewing are thought to be regulated by a CPG.

The main characteristics of CPGs rhythms are coordination, variety, sensory feedback and adaptability to the environment. For example: the chirping of a cricket is periodic most of the time but also has corrections in time, the locomotion of horses exhibits only specific gaits: walking, trotting, canter and galloping, having patterns of four, three or two beats per cycle. Human rhythm production, even when mediated by more sophisticated and distributed neural processes, probably also relies on neural oscillators interacting and resonating with rhythmic stimuli [2]. We take inspiration from these biological rhythmic networks for building the core of the SANTIAGO interface, capable of producing complex and adaptable rhythmic patterns.

3. SANTIAGO'S INTERFACE

3.1 Architecture

Santiago's interface is built upon a modular structure, exhibited in figure 1. Both back and front-ends are built in PureData as a set of hierarchical organized abstractions allowing, through dynamic patching, the rapid creation, interconnection and elimination of units.

A wide range of inputs may be handled according to the user's needs, and routed to virtually any parameter of the network; for this reason Santiago is also suitable for its use with performance controllers and software environments with whom the users may already have some practice or developed performance skills. Continuous data can be scaled to best fit the biological units, and can be mapped to input currents of the neurons or synaptic weights of the network. Event data can be used to excite or inhibit a neuron with a current pulse and can also change the neuron firing mode, or type (see fig. 3).

The network is the core of the system, as it generates the complex behavior in time; its outputs are spikes (events) and firing rates (continuous) from all neurons. Those out-

puts can be sent to diverse internal sonification and visualization modules or, via OSC, MIDI and proper outputs, to other devices such as external samplers and synthesizers, sequencers or even actuators and audiovisual engines. The firing rates are particularly useful since they can control many parameters at once (for instance, amplitude, pitch, duration, etc.) even using different mapping curves for each parameter.

3.2 Design and user experience

When the users load Santiago, the main panel shows up 2, presenting a visual interface designed to provide a rapid intuition of the environments functionality. When clicked, the labeled buttons grouped accordingly open the corresponding panels, while giving the user a visual feedback by changing their colors from default grey to default or customizable colors.

The GUI is intended to be simple, intuitive, versatile and highly configurable. Many design decisions have been taken in the pursuit of a consistent usable interface. Depicted in the panels shown for the first example in the following section, many of these decisions are:

- Each neuron is identified with a reference block at the left of the NEURONS, NETWORK and EVENT VIEW panels, containing its identification number and, in the first two panels, a type status button from which the user can set the neuron to E or I for Excitatory or Inhibitory type.

- The identification number background blinks with a configurable color, by default black, every time the neuron generates a spike, therefore providing instant visual feedback of its activity. For no activity, the default color is grey.

- Sizes of NEURONS, NETWORK and EVENT VIEW modules are consistent to each other, so aligning those panels is a comfortable way to design and visualize neural activity and interconnection.

- The interface colors allow the user to quickly grasp its functioning principles, even having the possibility to set them individually and save them into GUI presets. For instance, if a neuron has an excitatory behavior, the type button in its reference block will have the same color than all its synaptic connections in the NETWORK panel.

- Every panel includes a global module configuration located at the top, by default with a pale green background, from which the user may quickly configure several modules at once, or large amounts of values in parallel, using the keyboard or mouse.

- The modules offer a consistent preset loading and saving submodule, that allows handling up to 10 presets each, and infinite presets banks.

Currently, the connectivity matrix found in the NETWORK panel, shows a random button that sets different values for each synapse weight and turns neurons into E or I types, thus inviting the users to explore the possibilities of that parameter space with a single click. Beyond this explorations, if the user is in the search of precise results, we encourage conscious design of the networks. In the next section we show some examples of outputs generated by simple designed networks.

3.3 Generating rhythms

The NEURONS module implements the model described by Eqs. (1.a - 1.c), and allows the user to control the four parameters and the inputs. For the sake of simplicity six prototypical neuron types are also available as presets: Regular Spiking (RS), Intrinsically Bursting (IB), Chattering (CH), Low Threshold (LT), Fast Spiking (FS) and Resonator (RZ). Representative patterns of spikes for three of these presets are displayed in Fig. 3. As with the rest of the modules, the user is invited to explore the parameter space, design his own configurations and save them into personal presets.

The NETWORK module allows the user to establish the synapses, or connections between neurons, selecting the intensity and the delay of inhibitory or excitatory action. In general terms, a stronger synapse intensity will produce a quicker excitatory or inhibitory action upon the neurons that receive the current pulse, and a weaker value will produce a more delayed action upon the activity of the network. Also lower currents above the firing threshold of the neurons tend to exhibit a clearer, more regular and spaced rhythmic activity than higher currents, more suitable for granular synthesis or more statistically perceived occurrences of the events.

In order to illustrate how the neural network core of SANTIAGO can generate a wide diversity of rhythmic patterns using few units we choose three examples that are both biologically inspired and musically interesting.

Our first example comprises only two neurons and is a good illustration of how the intrinsic dynamics of each units can interact in its simplest expression: a pair of excitatory and inhibitory neurons.

In Fig 4 we show how to construct this simple network and visualize its output. The NEURONS panel (A) show the list of numbered neurons. For each units it is possible to adjust the DC current, and neuron type using the individual parameters or the presets. The NETWORK panel (B) displays the connectivity matrix of the neural network. This allows to adjust the synaptic weights, and allows a quick view to the inhibitory (red) or excitatory (blue) condition of the neurons. The EVENT VIEW panel (C) show the events in real time.

In this case, the first neuron (N_1) is a inhibitory LT, which normally has a regular spiking pattern. The second neuron (N_2) is an excitatory CH with a strong input current. When N_2 fires a discharge it excites N_1 and eventually a spike occurs in the this neuron, that in turn inhibits N_2, interrupting the discharge pattern. A pattern of three spikes of N_2 and one spike of N_1 in response is clearly recurrent, but the exact relative timing of the events is not the same, and also long burst of N_2 activity alternates with this pattern. Slight changes in N_2 DC current provokes drastic changes in this rhythmic pattern.

Our second example illustrates synchronization. Three inhibitory neurons (LT) in a ring (N_1 inhibits N_2 and N_2 inhibits N_3, which in turn inhibits N_1) stimulated by a DC current are a paradigmatic example of rhythmic behavior. The three units alternate their spikes in a cycle, and never two spikes occur simultaneously due to the inhibition. The relative phase of the spikes within the pattern depends on the initial state. In our example a fourth slow CH excitatory neuron force the three units to fire in phase, overcoming inhibition. When the burst of the CH neuron ends the inhibitory ring starts its cycle again, but with different relative phases. An EVENT VIEW of this network is displayed in Fig. 5.

Our last example (Fig . 6) is built with four neurons: N_1 is the only one with DC input current, and excitates N_2

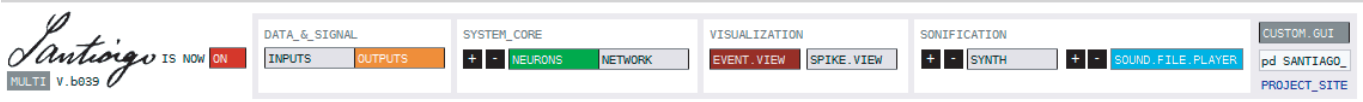


Figure 2: Main panel of Santiago. An intuitive GUI for accessing the system's features and panels. Dynamic patching allows creating and erasing panel modules by simply clicking the (+) or (-) black backgrounded buttons. This occurs interdependently for NEURONS, NETWORK, EVENT VIEW and SPIKE VIEW minimal units, which are created an labeled automatically for further use; and independently for each internal sonification module. The patch underlying programming is accessible at the right and is available for customization, as well as GUI colors. At the bottom-right corner, there is a link to the project's website, containing examples, documentation, news and updated versions of the interface.

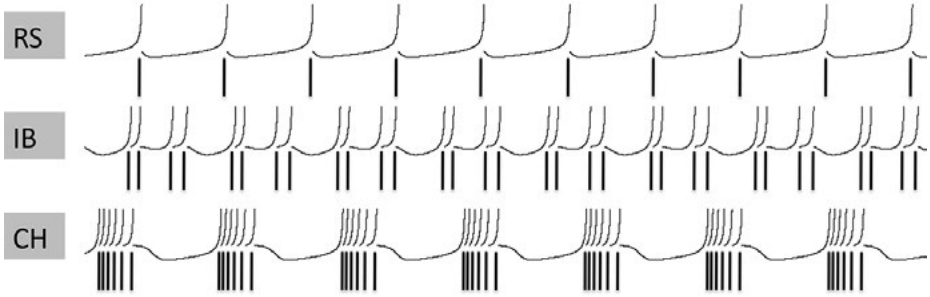


Figure 3: The figure depicts the visualization output of the SPIKE VIEW, for three prototypical behaviors of the neuron model of choice. Regular Spiking (RS), Intrinsecally Bursting (IB), Chattering (CH). Below each trace a vertical bar indicates the occurrence of a spike, as in the EVENT VIEW panel.

and N_4 with a synapse weight of 43. N_2 is also excitatory and has a synapse weight of 33 with N_3, which inhibits N_4 with a synapse weight of 33. It is interesting to notice the variety of rhythm patterns, polyrhythms and time signature changes that occur only by changing the input DC currents for N_1 while mantaining the exact same network configuration. Network outputs for input DC current values for N_1 of 7, 11, 14 and 19 are depicted y A, B, C and D respectively.

4. PERFORMANCE SCENARIOS

Santiago can assume different roles in composition processes and performance scenarios, ranging from an interactive tool for the generation of materials usable in deferred time compositions -for electronic and acoustic instruments-, to a real-time complete performance suite. In any case, since the creation of units for each module is highly simplified avoiding time consuming manual patching, the user focuses only in artistic and not programming issues, designing the network behavior and sonificating it almost instantaneously. One of the most powerful features in Santiago, specially for its real-time usage in a live performance, is the fact that the user can change all the parameters on-the-fly, without restarting the simulation. Also a very practical preset management has been implemented at different levels of hierarchy. Not only presets can be handled for individual modules and general modules, but also global scenes can be saved and loaded. This permits multiple parameters for all panels to be modified at once, just by pressing a button. Finally, the user can input an interpolation time between loaded presets, producing unexpected transitions between expected states.

Depending on the artistic requirements and, of course, the available equipment, a single computer may be sufficient for every performance aspect. When more processing power is needed, the artist can make use the OSC input/output capabilities and work with multiple computers running Santiago in a LAN, each one for a different function. For instance, one handling the input data processing and running the system core, and another one for sonification and visualization. In this sense, multi-user collaborative performance in-place with single or multiple computers is, of course, also an option. OSC communication also opens the field for telematic performances with Santiago. For example, the neurons of a big network could be distributed on several machines. Here, also the variable of net delay times is introduced into the system.

5. CONCLUSIONS

We present a novel approach for biological interactive systems based on realistic neural models with special focus on rhythm and generative music. This environment was designed for real time performance in a single or multiple computers. Also, the whole software implementation structure is modular, allowing easy mapping of external signals to internal parameters and internal signals to media outputs. It consist in a core, where the dynamical system operates, and a collection of modules for standard audio and data operations that interact with the core.

While mapping strategies are still in development, the core is fully functional and allows a wide range of rhythms and textures. In future versions the biological core will have a plasticity module to include dynamic changes in the synaptic weights depending on some learning rule.

The visualization is still in development and in the future will include a visual interface to set up the network and neurons geometrically. As for the visual outputs, the event view will be extended to include more visual attributes as for example, intensity or pitch. This will be useful for interaction with instrument players as a real time score output.

In the sonification direction, a spatialization tool will be included in concordance with spatial positions of the neurons in the visual input.

For more references and audible examples go to `http:`

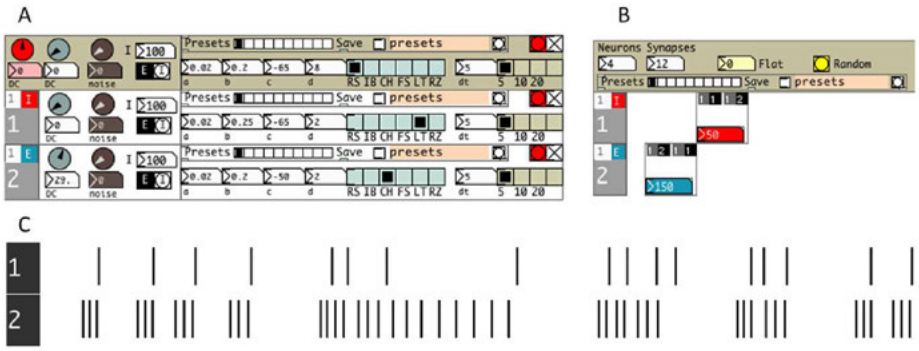


Figure 4: NEURONS panel (A), NETWORK panel (B) and EVENT VIEW (C) of a simple two-neuron example built in SANTIAGO. The excitatory-inhibitory pair produces a non-recurrent pattern of dishcarges.

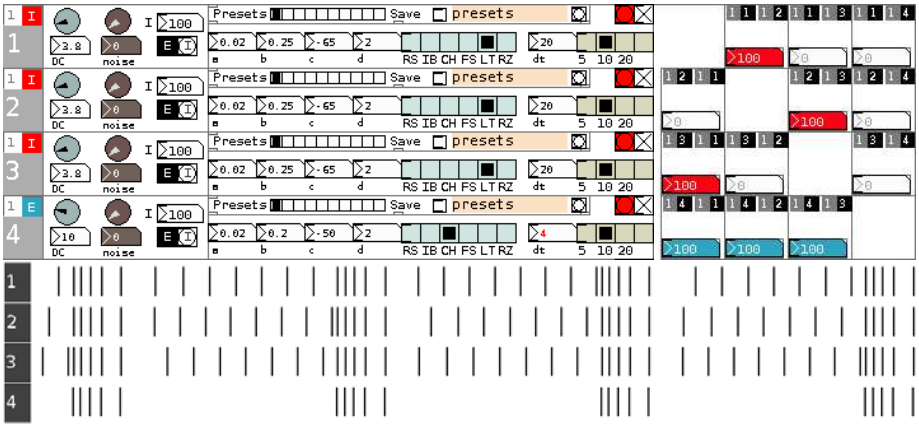


Figure 5: A simple CPG built with four neurons. A group of three neurons that inhibit one neighbor and one excitatory neuron that synchronizes them. An alternating cycle begins in the inhibitory ring, after each synchronization, that evolves into a regular inter-event time.

`//lapso.org/santiago`

6. ACKNOWLEDGMENTS

This work was done with partial support from CONICET.

7. REFERENCES

[1] G. Buzsáki. *Rhythms of the Brain*. Oxford University Press, USA, 2006.

[2] J. Grahn and M. Brett. Rhythm and beat perception in motor areas of the brain. *Journal of Cognitive Neuroscience*, 19(5):893–906, 2007.

[3] E. Izhikevich. Simple model of spiking neurons. *Neural Networks, IEEE Transactions on*, 14(6):1569–1572, 2003.

[4] H. Kerllenevich, P. Riera, and M. Eguia. SANTIAGO - A Real-time Biological Neural Network Environment for Generative Music Creation. *EvoApplications 2011, Part II, LNCS*, 6625:344–353, 2010.

[5] E. Marder and D. Bucher. Understanding circuit dynamics using the stomatogastric nervous system of lobsters and crabs. *Annual review of physiology*, 69:291, 2007.

[6] E. Miranda and J. Matthias. Granular sampling using a pulse-coupled network of spiking neurons. *Applications on Evolutionary Computing*, pages 539–544, 2005.

[7] G. Shepherd. *The synaptic organization of the brain*. Oxford University Press, USA, 2004.

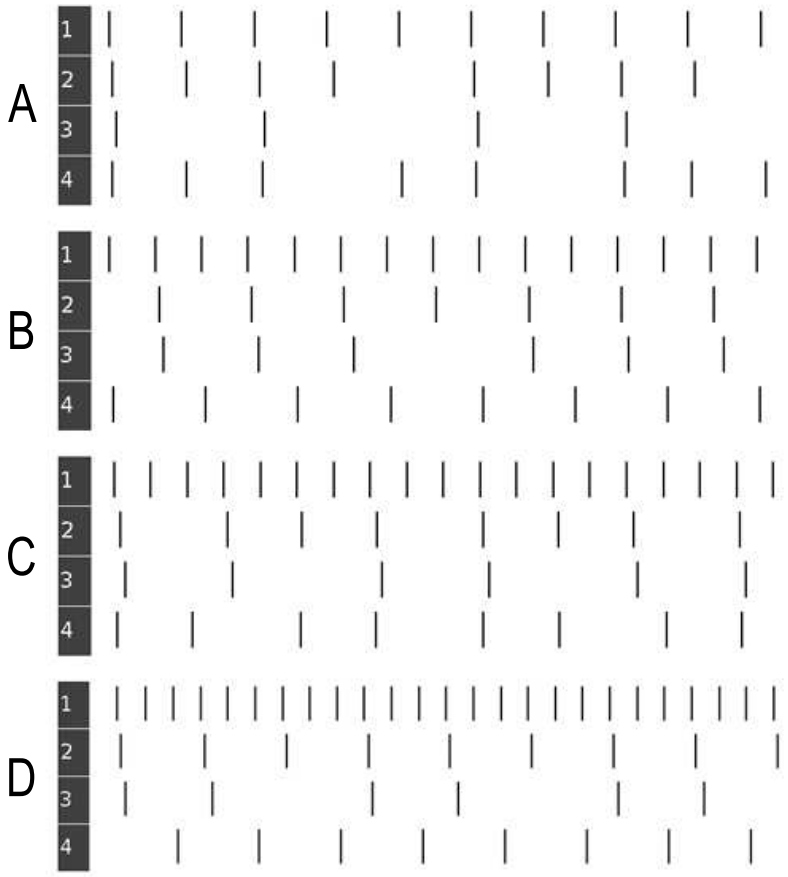


Figure 6: In A, a network output is shown for an Input current for $N_1 = 7$. The result for N_1, N_2 and N_3, is a 5/8 pattern, in which N_1 goes in eights, N_2 repeats a pattern of 4 eights + 1 silence, and N_3 stresses 2+3 groups. N_4 alternates statistically 3 eights + 1 silence and 2 eights + silence

Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping

Nicholas Gillian
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
ngillian01@qub.ac.uk

R. Benjamin Knapp
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
b.knapp@qub.ac.uk

Sile O'Modhrain
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
sile@qub.ac.uk

ABSTRACT

This paper presents a novel algorithm that has been specifically designed for the recognition of multivariate temporal musical gestures. The algorithm is based on Dynamic Time Warping and has been extended to classify any N-dimensional signal, automatically compute a classification threshold to reject any data that is not a valid gesture and be quickly trained with a low number of training examples. The algorithm is evaluated using a database of 10 temporal gestures performed by 10 participants achieving an average cross-validation result of 99%.

Keywords

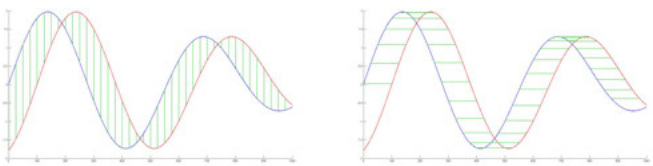
Dynamic Time Warping, Gesture Recognition, Musician-Computer Interaction, Multivariate Temporal Gestures

1. INTRODUCTION

Musicians commonly use body movements such as hand, arm and head gestures to communicate with other performers live on stage. This method of interaction is still difficult, however, between a musician and a computer despite the accessibility of cheap sensor devices and flexible machine learning software that can be used to recognise such gestures. Musical gestures can be difficult for a computer to recognise because many gestures are not simply static postures but consist of a cohesive sequence of movements that occur over a variable time period. Further, these temporal gestures commonly require multiple sensors to adequately capture the movement and a computer must therefore construct a model that describes not only the relationship between all the sensors at time t, but also how this relationship changes over time. Training a computer to automatically recognise *musical* temporal gestures also creates a number of interesting challenges that are not commonly found in other areas of human-computer interaction (HCI). This is because a musician will frequently want to use their own sensor technology to capture gestures that are inherently personal to that one performer; using the recognition of these gestures to interact with a specific piece of real-time audio performance software. The algorithms used to recognise a performer's gestures cannot therefore, in many instances, be pre-trained prior to being distributed to a musician; but must instead be trained by the musician. A musician therefore requires a recognition algorithm that can be quickly trained with a few examples of the performer's gestures, captured by whatever sensor is most applicable for that performer. The recognition algorithm employed to classify such gestures should, therefore, not be constrained to only recognise the gestures captured by one specific sensor, such as an accelerometer or webcam, but should work with any N-dimensional temporal signal. The key concept about designing and evaluating such an algorithm for the recognition of musical gestures is that, unlike many other areas of machine learning, the goal of the algorithm should be to achieve a low intra-personal generalisation error for the one user that trained the algorithm as opposed to a low inter-personal generalisation error. This paper presents an algorithm that has been specifically designed for the recognition of temporal musical gestures. The algorithm is based on *Dynamic Time Warping* (DTW) and has been extended to classify any N-dimensional, also known as multivariate, signal, automatically compute a classification threshold to reject any data that is not a valid gesture and be quickly trained with a low number of training examples.

2. DYNAMIC TIME WARPING

Dynamic Time Warping is an algorithm that can compute the similarity between two time-series, even if the lengths of the time-series do not match. One of the main issues with using a distance measure (such as Euclidean distance) to measure the similarity between two time-series is that the results can sometimes be very unintuitive. If for example, two time-series are identical, but slightly out of phase with each other, then a distance measure such as the Euclidean distance will give a very poor similarity measure. Figure 1 illustrates this problem. DTW overcomes this limitation by ignoring both local and global shifts in the time dimension [13].



(a) Euclidean Distance (b) Dynamic Time Warping

Figure 1: Two identical time-series, slightly out of phase with each other, matched using Euclidean distance and Dynamic Time Warping

2.1 Related Work

There has been much work over the last two decades in applying DTW to such varying fields as database indexing

[6] [1], handwriting recognition [17] and gesture recognition [2] [4]. The vast majority of the recent work into DTW has focused on making the algorithm more computationally efficient [6] [7], with the time series in these works all being uni-dimensional signals. Proposed improvements to DTW included constraining the warping path [12] [5], lower-bounding [8] [10], numerosity reduction [19] and recursive resolution projection [13]. It has only been in recent years that research has been conducted into extending DTW to multiple dimensions, with the exception of the early work by Stettiner [14] who proposed an extension of DTW to multiple dimensions for the application of speech recognition. Vlachos et. al. [16] extended DTW to match two-dimensional time series. In previous work by Holt et. al. [15] and also separately by Ko et. al. [9], multi-dimensional DTW was achieved by using a distance function such as the absolute sum, Euclidean distance or cosine correlation coefficient to compute the distance over all the dimensions in the test time series with a template time series for each sample in time. The result of this distance function was used by the standard DTW algorithm to compute the warping cost between the test time series and the template time series. Wollmer et. al. [18] proposed a different approach to multi-dimensional DTW, using a three-dimensional distance matrix to compute the minimum distance between the input time series and a reference time series. This work used a bimodal input signal (speech data and gesture data captured by a mouse) and would therefore be computationally expensive to expand to an N-dimensional input stream as a large dimensional space would need to be constructed and navigated for each of the G gestures in the database.

Merrill et al. [11] successfully applied DTW to the recognition of musical gestures. Using the custom-built FlexiGesture (a two handed device that featured a number of sensors including accelerometers, gyroscopes, along with squeezing, bending and twisting sensors), a user could train the system to recognise up to 10 temporal gestures by pressing a 'trigger' button which started the data recording process, releasing the button when the gesture was completed. The system then asked the user to continually re-perform the gesture as it trained a template model for that gesture. Tests showed that the system was able to classify novel gestures into one of 10 classes with up to 98% accuracy.

2.2 One-Dimensional DTW

The foundation algorithm for DTW is as follows. Given two, one-dimensional, time-series, $\mathbf{x} = \{x_1, x_2, ..., x_{|\mathbf{x}|}\}^\intercal$ and $\mathbf{y} = \{y_1, y_2, ..., y_{|\mathbf{y}|}\}^\intercal$, with respective lengths $|\mathbf{x}|$ and $|\mathbf{y}|$, construct a *warping path* $\mathbf{w} = \{w_1, w_2, ..., w_{|\mathbf{w}|}\}^\intercal$ so that $|\mathbf{w}|$, the length of $\mathbf{w}$ is:

$$\max\{|\mathbf{x}|, |\mathbf{y}|\} \leq |\mathbf{w}| < |\mathbf{x}| + |\mathbf{y}| \tag{1}$$

where the kth value of $\mathbf{w}$ is given by:

$$\mathbf{w}_k = (\mathbf{x}_i, \mathbf{y}_j) \tag{2}$$

A number of constraints are placed on the warping path, which are as follows:

- The warping path must start at: $\mathbf{w}_1 = (1, 1)$
- The warping path must end at: $\mathbf{w}_{|\mathbf{w}|} = (|\mathbf{x}|, |\mathbf{y}|)$
- The warping path must be continuous, i.e. if $\mathbf{w}_k = (i, j)$ then $\mathbf{w}_{k+1}$ must equal either (i, j), $(i + 1, j)$, $(i, j + 1)$ or $(i + 1, j + 1)$
- The warping path must exhibit monotonic behavior, i.e. the warping path can not move backwards

There are exponentially many warping paths that satisfy the above conditions. However, we are only interested in finding the warping path that minimizes the normalised total warping cost given by:

$$\min \quad \frac{1}{|\mathbf{w}|} \sum_{k=1}^{|\mathbf{w}|} DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j}) \tag{3}$$

where $DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j})$ is the distance function (typically Euclidean) between point i in time-series $\mathbf{x}$ and point j in time-series $\mathbf{y}$, given by $\mathbf{w}_k$. The minimum total warping path can be found by using dynamic programming to fill a two-dimensional ($|\mathbf{x}|$ by $|\mathbf{y}|$) cost matrix $\mathbf{C}$. Each cell in the cost matrix represents the accumulated minimum warping cost so far in the warping between the time-series $\mathbf{x}$ and $\mathbf{y}$ up to the position of that cell. The value in the cell at $\mathbf{C}_{(i,j)}$ is therefore given by:

$$\mathbf{C}_{(i,j)} = DIST(i, j) + \min\{\mathbf{C}_{(i-1,j)}, \mathbf{C}_{(i,j-1)}, \mathbf{C}_{(i-1,j-1)}\} \tag{4}$$

which is the distance between point i in the time-series $\mathbf{x}$ and point j in the time-series $\mathbf{y}$, plus the minimum accumulated distance from the three previous cells that neighbor the cell i,j (the cell above it, the cell to its left and the cell at its diagonal). When the cost matrix has been filled, the minimum possible warping path can easily be calculated by navigating through the cost matrix in reverse order, starting at $\mathbf{C}_{(|\mathbf{x}|,|\mathbf{y}|)}$, until cell $\mathbf{C}_{(1,1)}$ has been reached, as illustrated in Figure 2. At each step, the cells to the left, above and diagonally of the current cell are searched to find the minimum value. The cell with the minimum value is then moved to and the previous three cell search is repeated until $\mathbf{C}_{(1,1)}$ has been reached. The warping path then gives the minimum normalised total warping distance between $\mathbf{x}$ and $\mathbf{y}$:

$$DTW(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{w}|} \sum_{k=1}^{|\mathbf{w}|} DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j}) \tag{5}$$

Here, $\frac{1}{|\mathbf{w}|}$ is used as a normalisation factor to allow the comparison of warping paths of varying lengths.

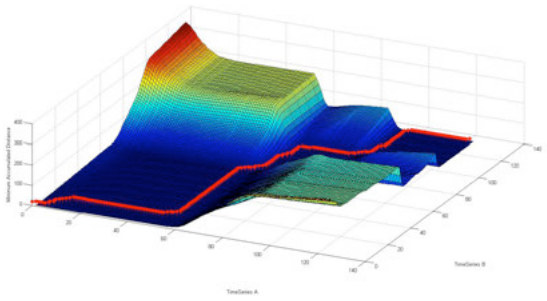


Figure 2: Cost Matrix and the Minimum Warp Path through it (indicated by the red line)

2.3 Numerosity Reduction

DTW is a useful tool for computing the distance between two time-series. It is, however, a computational costly algorithm to use for real-time recognition, as every value in the cost matrix must be filled. Clearly this is unusable for real-time recognition purposes, particularly if the unknown time-series is being matched against a large database of gestures. To speed up both the training of the gesture templates and the real-time classification of an unknown N-dimensional input time-series, we tested various methods of numerosity

reduction. Perhaps one of the most rudimentary methods for numerosity reduction is to downsample the time-series by a factor of n. To avoid aliasing, the data is filtered using a low-pass FIR filter with a rectangular window and a filter order of n.

2.4 Constraining the Warping Path

Another method commonly adopted for improving the efficiency of DTW is to constrain the warping path so that the maximum warping path allowed cannot drift too far from the diagonal. Controlling the size of this warping window will greatly affect the speed of the DTW computation. If the warping window is small, a large proportion of the cost matrix does not need to be searched or even constructed. The size of the warping window can be controlled by varying the parameter r, given as the percentage of the length of the template time-series. The warping window is then set as the distance, r, from the diagonal to directly above and to the right of the diagonal. This type of global constraint is referred to as the Sakoe-Chiba band [12]. Itakura has also proposed another global constrained based on a parallelogram [5].

3. ND-DTW

Section 2.1 describes the standard implementation of DTW for two, uni-dimensional time-series. It is common, however, in computational fields such as gesture recognition to have time-series that feature multiple-dimensions, such as data captured by a 3-axis accelerometer. It is in this instance that we require an implementation of DTW that can compute the distance between two N-dimensional time-series. We will use the common approach used by [15][9] to compute the distance between two N-dimensional time-series. This takes the summation of distance errors between each dimension of an N-dimensional template and the new N-dimensional time-series. The total distance across all N dimensions is then used to construct the warping matrix $\mathbf{C}$. We will use the Euclidean distance as a distance measure across the N dimensions of the template and new time-series.

$$DIST(i,j) = \sqrt{\sum_{n=1}^{N}(i_n - j_n)^2} \tag{6}$$

The following section describes our N-Dimensional Dynamic Time Warping (**ND-DTW**) algorithm. In the training stage, an N-dimensional template (ϕ_g) and threshold value (τ_g) for each of the G gestures is computed. In the real-time prediction stage a new N-dimensional time-series is classified against the template that gives the minimum normalised total warping distance between the N-dimensional template and the unknown N-dimensional time-series. We will now discuss each element of the algorithm in detail.

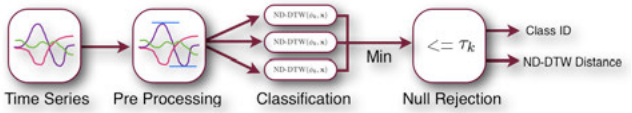


Figure 3: The ND-DTW classification chain

3.1 Training the ND-DTW Algorithm

In order for ND-DTW to be used as a real-time recognition algorithm, a template must first be created for each gesture that needs to be classified. A template can be computed by recording M_g training examples for each of the G gestures that are required to be recognised. After the training data has been recorded, each of the G templates can be found by computing the distance between each of the M_g training examples for the gth gesture and searching for the training example that provides the minimum normalised total warping distance when matched against the other M_g-1 training examples in that class. The gth template (ϕ_g) is therefore given by:

$$\phi_g = \arg\min_i \quad \frac{1}{M_g - 1}\sum_{j=1}^{M_g}\mathbf{1}\{\text{ND-DTW}(\mathbf{X}_i, \mathbf{X}_j)\} \qquad 1 \le i \le M_g \tag{7}$$

where the $\mathbf{1}\{\cdot\}$ that surrounds the ND-DTW function is the indicator bracket, giving 1 when $i \neq j$ or 0 otherwise and $\mathbf{X}_i$ and $\mathbf{X}_j$ are the ith and jth N-dimensional training examples for the gth gesture in the form of $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and $\mathbf{x} = \{x_1, x_2, ..., x_{|\mathbf{x}|}\}^\intercal$. The ND-DTW function in (7) is simply the extension of the standard DTW algorithm to N-dimensions:

$$\text{ND-DTW}(\mathbf{X}, \mathbf{Y}) = \quad \min \quad \frac{1}{|\mathbf{w}|}\sum_{k=1}^{|\mathbf{w}|} DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j})$$

$$DIST(i,j) = \sqrt{\sum_{n=1}^{N}(i_n - j_n)^2} \tag{8}$$

3.2 Multi-Threaded Training

One major advantage of using the DTW algorithm is that each template (i.e. each gesture) can be computed independently from the other templates. This is of particular use on new machines that feature multiple processors as a multi-threaded training approach can be adopted in which each template's training routine is launched in a separate thread. This training approach greatly speeds up the overall training time for a DTW classification system as one template does not need to wait for the previous template to be trained before it can start its own training routine.

The DTW algorithm also has one other advantage in that, if a new gesture is added to an existing trained model or an existing gesture is removed, the entire model does not need to be retrained. Instead, a new template and threshold value only needs to be trained for the new gesture, thus greatly reducing the training time. If an existing gesture is removed from the model then no re-training is required as the DTW classification system simply removes this template and threshold value from its 'database'. This is not the case for other machine learning algorithms, such as an Artificial Neural Network, as the entire system would need to be retrained from scratch any time a new gesture is added or removed.

3.3 Classification Using ND-DTW

After the ND-DTW algorithm has been trained, an unknown N-dimensional time-series $\mathbf{X}$ can be classified by computing the normalised total warping distance between $\mathbf{X}$ and each of the G templates in the model. c, the classification index representing the gth gesture is then given by finding the corresponding template that gave the minimum normalised total warping distance:

$$c = \arg\min_g \quad \text{ND-DTW}(\phi_g, \mathbf{X}) \qquad 1 \le g \le G \tag{9}$$

3.4 Determining the Classification Threshold

Using equation (9) $\mathbf{X}$, an unknown N-dimensional time-series, can be classified by calculating the distance between it and all the templates in the model. The unknown time-series $\mathbf{X}$ can then be classified against the template that results in the lowest normalised total warping distance. This method will, however, give false positives if the N-dimensional input time-series $\mathbf{X}$ is in-fact not made up of any of the gestures in the model. This false classification problem can be mitigated by determining a classification threshold for each template gesture during the training phase. In the prediction phase, a gesture will only be classified against the template that results in the lowest normalised total warping distance, if this distance is less than or equal to the gesture's classification threshold. If the distance is above the classification threshold, then the algorithm will classify the gesture against a null class, indicating that no match was found:

$$\hat{c} = \begin{cases} c & if(d \leq \tau_g) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where c is given by equation (9), d is the total normalised warping distance between ϕ_g and $\mathbf{X}$ and τ_g is the classification threshold for the gth template.

The classification threshold for each template can be set as the average total normalised warping distance between ϕ_g and the other $M_g - 1$ training examples for that gesture, plus γ standard deviations:

$$\tau_g = \mu_g + (\sigma_g \gamma) \tag{11}$$

where

$$\mu_g = \frac{1}{M_g - 1} \sum_{i=1}^{M_g} \mathbf{1}\{\text{ND-DTW}(\phi_g, \mathbf{X}_i)\} \tag{12}$$

$$\sigma_g = \sqrt{\frac{1}{M_g - 2} \sum_{i=1}^{M_g} \mathbf{1}\{(\text{ND-DTW}(\phi_g, \mathbf{X}_i) - \mu_g)^2\}} \tag{13}$$

where the $\mathbf{1}\{\cdot\}$ that surrounds the ND-DTW function is the indicator bracket, giving 1 when $i \neq$ the index of the training example that gave the minimum normalised total warping distance when matched against the other M_g-1 training examples in that class (i.e. the template) or 0 otherwise and $\mathbf{X}_i$ is the ith training example for the gth class. γ can be initially set to a number of standard deviations (e.g. 2) during the training phase and later adjusted by the user in the real-time predication phase until a suitable classification/rejection level has been achieved.

It is critical when calculating the classification threshold for each of the g gestures to perform any preprocessing such as scaling or downsampling in the same order as it would be performed during the real-time classification stage. If this is not completed in the same order then the optimal classification threshold will not be found. We will now discuss the various preprocessing options that can be used for ND-DTW.

3.5 Preprocessing for ND-DTW

Pre-processing is necessary for ND-DTW if either (a) any of the N-dimensional data originate from a different source range or (b) if invariance to spatial variability and variability of signal magnitude is desired. We now discuss both of these points and give appropriate preprocessing solutions for each.

3.5.1 Varying Input Source Ranges

It is important for each of the N-dimensional data in the time-series $\mathbf{X}$ to originate from a common source range. If this is not the case then one or more of the dimensions may heavily weight the results of the DTW. If each of the N-dimensional data do not originate from a common source range then each channel should be scaled using min-max normalisation prior to both the training of the templates and real-time prediction.

3.5.2 Invariance to Amplitude & Spatial Variability

Spatial variance and variability in the signal amplitude can be mitigated by first z-normalising both the input time-series and also the recognition templates. Z-normalisation will give both the input and template time-series zero mean and unit variance, therefore removing any affect that spatial variation or variability in the signal amplitude may have had. Keogh et. al. [7] also proposed using the derivative of the input signals to account for similar spatial problems. This method was also used successfully by Holt et. al. [15].

3.6 Real-time Implementation

The ND-DTW algorithm has been fully integrated into the SEC[1], a machine learning toolbox that has been specifically developed for musician-computer interaction [3]. The SEC is a third party toolbox consisting of a large number of machine learning algorithms that have been added to EyesWeb[2], a free open software platform that was established to support the development of real-time multimodal distributed interactive applications.

4. DTW EXPERIMENTS

Three experiments were run to validate the classification abilities of the ND-DTW algorithm. To test the algorithm 10 participants were recruited and asked to perform 25 repetitions of 10 gestures. The 10 gestures consisted of 'air drawing' several numbers and shapes with the right hand, including the numbers 1 -5, a square, a circle, a triangle, a horizontal line similar to a downbeat conducting gesture and a vertical line similar to a sidebeat conducting gesture. Each participant wore a Polhemus magnetic tracking sensor mounted on their right wrist which was sampled at 120Hz. The data collected from all 10 participants will be referred to as the numbers-shapes dataset. Because the ND-DTW algorithm has been specifically designed for the recognition of musical gestures, with the objective of creating an algorithm that can be quickly trained to accurately classify the musical gestures of the one performer that trained it, each experiment will validate the intra-personal generalisation abilities of the algorithm as opposed to the inter-personal generalisation.

4.0.1 Experiment A

This experiment tests the ND-DTW algorithm's ability to correctly classify the pre-segmented data from the numbers-shapes dataset. For each participant, a ND-DTW model was trained using 10-fold cross-validation, with the average cross-validation ratio (**ACVR**) taken over all 10 participants being used to evaluate the algorithm. This experiment was run with four conditions (C1) scaling off, z-normalisation off; (C2) scaling on, z-normalisation off; (C3) scaling off, z-normalisation on and (C4) scaling on z-normalisation on. γ was set to 2 and a downsample factor of 5 was used for all conditions. Condition C2 achieved the

[1] http://www.somasa.qub.ac.uk/ ngillian/SEC.html
[2] http://musart.dist.unige.it/EywMain.html

maximum ACVR of 99.37%, however the other conditions also achieved excellent classification results of 98.85% for C1, 98.95% for C3 and 99.37% for C4. This test shows that the ND-DTW algorithm provides excellent classification results on pre-segmented data. The ND-DTW algorithm achieved a perfect recognition result of 100% for several participants, with the algorithm achieving a classification result of over 99% for all but 1 participant. Figure 4 shows the cross-validation results for each of the 10 participants.

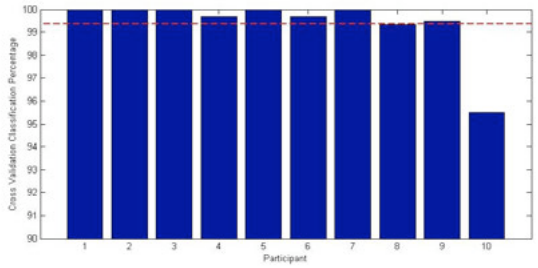


Figure 4: The cross-validation classification results for each of the 10 participants in condition C2. The ACVR is illustrated by the dotted horizontal line

4.0.2 *Experiment B*

This experiment tests the classification abilities of the ND-DTW algorithm with respect to a minimal amount of training data. This is an important test for music as, if a model can achieve as good a classification result with 2 training examples as it can with 20 training examples, then a performer can save time in both collecting the training data and also in training the model. For each participant, a ND-DTW model was trained using η randomly selected training examples from each of the 10 gestures and tested with the remaining data. η ranged from 3 - 20, starting at 3 as opposed to 1 because at least 3 training examples are required to estimate the threshold value for each template and stopping at 20 to allow at least 5 test examples per trial. To ensure that the results of this test were not weighted by a 'lucky' random selection of the best template from the 25 training samples of each gesture, each test for η was repeated 10 times and the average correct classification ratio (**ACCR**) was recorded and used for validation of the algorithm. γ was set to 2 for this experiment and a downsample factor of 5 was used. Figure 5 shows the ACCR for each iteration of η. This test shows that the number of training examples significantly effects the classification abilities of the ND-DTW algorithm. The ND-DTW algorithm achieved a moderate ACCR value of 74.74% with just 3 training examples. With 20 training examples it was able to achieve an ACCR value of 92.19%. It should be noted that the standard deviation over each iteration of η and across all 10 participants was very high. This shows that the classification abilities of the ND-DTW algorithm is heavily dependent on getting 'the best' training examples. Several participants, for example, achieved an ACCR value of $> 90\%$ with just 3 training examples. The same participants, however, also achieved an ACCR value of $< 70\%$ with the same number of training examples, showing that the 'quality' of the training examples heavily influences the results of the classification algorithm. The results of this test suggest that at least 11 training examples are required per-gesture if the user wants to achieve a robust classification result of $> 90\%$.

4.0.3 *Experiment C*

This experiment tests the ND-DTW algorithm's ability to correctly classify data from the numbers-shapes dataset in

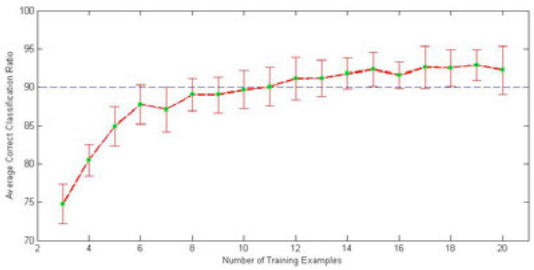


Figure 5: The ACCR values averaged across all 10 participants for each iteration of η. The horizontal blue line indicates the minimal training examples required to achieve a classification result of $> 90\%$

a continuous stream of data that also contains a number of null gestures. This evaluates two important aspects of the ND-DTW algorithm for the recognition of multivariate temporal gestures. Namely the algorithm's ability to correctly classify a set of temporal gestures from a continuous stream of data and also the algorithm's ability to reject any null gesture that is not contained in the model's database.

For each participant, a ND-DTW model was trained using 12 randomly selected training examples from each of the 10 gestures. After each model had been trained it was tested using a continuous stream of data. The continuous stream of data originated from the data-collection phase of the numbers-gestures database and contains all of the participant's trial recordings. The continuous stream therefore contains not only all of the 25 gestures the participant performed (12 of which were used to train the model) but also, importantly, the participant's movements in between each trial along with the periods of rest.

The continuous stream was tested by running a sliding window of size w over the data stream in increments of 10. The window size, w, was individually calculated for each participant by taking the average length of the 10 ND-DTW templates for that participant. For the majority of the participants, w was 304, with the shortest window length of 248 and the longest window length of 368. At each increment, the data within the window was given to the ND- DTW model for classification. Each sample of data had been labelled with an ID tag (0 for a null-gesture or the gth class ID for an actual gesture). This ID tag was used to evaluate if the ND-DTW model had made the correct classification for each window of data. As some windows covered a section of data that contained half a gesture and noise, the classification results of a window were only counted if the maximum ID count within the window was greater than 80% of the length of the window. This test was evaluated using the average correct classification ratio (**ACCR**) given by the total number of counted correctly classified windows over the total number of counted windows. The average precision ratio (**APR**), average recall ratio (**ARR**) and average null recall ratio (**ANRR**) were also computed. These provided an indication of the exactness of the classifier for each gesture across all the participants ignoring the null gestures (APR), an indication of the performance of the classifier over a specific gesture across all participants ignoring the null gestures (ARR) and an indication of the performance of the classifier at correctly rejecting the null gestures (ANRR). γ was set to 5 and a downsample factor of 5 was used for this experiment.

This test was run with the same four conditions found in experiment A. The ACCR values for each of the four conditions were 83.31%, 84.18%, 74.15% and 74.15% respectively. Condition C2 with scaling on - z-normalisation off achieved the highest ACCR value of 84.18%. The max-

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
APR	0.91	0.89	0.80	0.79	0.92	0.95	0.83	0.96	0.95	0.96
ARR	0.93	0.78	0.85	0.81	0.82	0.66	0.94	0.93	0.92	0.90

Table 1: The average precision ratio (APR) and average recall ratio (ARR) for each gesture in condition C2

imum individual correct classification result of 95.23% was achieved by the algorithm for participant 1, while the algorithm achieved the minimum individual correct classification result of 64.09% for participant 8. Table 1 shows the APR and ARR results for condition C2, averaged over all 10 participants. The APR and ARR results show that the majority of classification errors were made by in the recall of the algorithm, as opposed to the precision of the algorithm. This shows that the ND-DTW algorithm made the majority of classification errors by misclassifying gesture i as a null gesture, rather than misclassifying gesture i as gesture j. The ANRR value of 0.88 indicates that the algorithm was successful at distinguishing a null-gesture from a gesture in the database 88% of the time.

These results suggest that the ND-DTW algorithm performed well at rejecting null gestures and also performed well at not misclassifying gesture i as gesture j. The main error that the ND-DTW algorithm made was in misclassifying gesture i as a null gesture. Increasing φ would have increased the threshold value for each gesture and therefore less gestures may have been misclassified as a null gesture. However, increasing this threshold value would have also increased the number of false-positive classifications (were a null gesture was falsely classified as gesture i). This problem illustrates the compromise that a user must make about the sensitivity of their classification system. Increasing the thresholding value will increase the likelihood that a gesture will be classified but it will also unfortunately increase the likelihood of false-positive misclassifications. It is for this specific reason that we have initially set the algorithm to calculate the threshold value as the mean plus two standard deviations of the error between the template and the remaining training examples for each gesture. The performer is then able to manually adjust this threshold value during the real-time 'live' prediction phase until the algorithm has reached a satisfactory recognition rate.

5. CONCLUSION

This paper has presented the ND-DTW algorithm which has been specifically designed for the recognition of multivariate temporal musical gestures. Three experiments have validated the algorithms ability to correctly classify a set of multivariate temporal gestures with a limited number of training examples and from a continuous stream of data that also contains null-gestures.

6. REFERENCES

[1] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[2] K. Forbes and E. Fiume. An efficient search algorithm for motion data using weighted pca. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, page 76. ACM, 2005.

[3] N. Gillian, R. B. Knapp, and S. O'Modhrain. A machine learning toolbox for musician computer interaction. In *NIME11*, 2011.

[4] A. Heloir, N. Courty, S. Gibet, and F. Multon. Temporal alignment of communicative gesture sequences. *Computer Animation and Virtual Worlds*, 17(3-4):347, 2006.

[5] F. Itakura. Minimum prediction residual principle applied to speech recognition. *Readings in speech recognition*, page 154, 1990.

[6] E. Keogh and M. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289. ACM, 2000.

[7] E. Keogh and M. Pazzani. Derivative dynamic time warping. In *First SIAM international conference on data mining*. Citeseer, 2001.

[8] E. Keogh and C. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[9] M. Ko, G. West, S. Venkatesh, and M. Kumar. Using dynamic time warping for online temporal fusion in multisensor systems. *Information Fusion*, 9(3):370–388, 2008.

[10] D. Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition*, 42(9):2169–2180, 2009.

[11] D. J. Merrill and J. A. Paradiso. Personalization, expressivity, and learnability of an implicit mapping strategy for physical interfaces. *CHI2005*, 2005.

[12] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Readings in speech recognition*, page 159, 1990.

[13] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[14] Y. Stettiner, D. Malah, and D. Chazan. Dynamic time warping with path control and non-local cost. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 1994.

[15] G. ten Holt, M. Reinders, and E. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In *Thirteenth annual conference of the Advanced School for Computing and Imaging*, 2007.

[16] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. *Proceedings of the 9th ACM SIGKDD int. conf. on Knowledge discovery and data mining*, 2003.

[17] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas. Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters. *International Journal on Document Analysis and Recognition*, 3:150–159, 2001.

[18] M. Wullmer, M. Al-Hames, F. Eyben, B. Schuller, and G. Rigoll. A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. *Neurocomputing*, 2009.

[19] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*, page 1040. ACM, 2006.

A Machine Learning Toolbox For Musician Computer Interaction

Nicholas Gillian
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
ngillian01@qub.ac.uk

R. Benjamin Knapp
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
b.knapp@qub.ac.uk

Sile O'Modhrain
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
sile@qub.ac.uk

ABSTRACT

This paper presents the SARC EyesWeb Catalog, (**SEC**), a machine learning toolbox that has been specifically developed for musician-computer interaction. The SEC features a large number of machine learning algorithms that can be used in real-time to recognise static postures, perform regression and classify multivariate temporal gestures. The algorithms within the toolbox have been designed to work with any N-dimensional signal and can be quickly trained with a small number of training examples. We also provide the motivation for the algorithms used for the recognition of musical gestures to achieve a low intra-personal generalisation error, as opposed to the inter-personal generalisation error that is more common in other areas of human-computer interaction.

Keywords

Machine learning, gesture recognition, musician-computer interaction, SEC

1. INTRODUCTION

It has long been the goal of many composers, performers and researchers alike to be able to use their own body movements to trigger, control and manipulate electronic sounds in real-time, live on stage. This goal is slowly being made possible by the ever decreasing cost of sensor devices, such as the Wii[1] or SHAKE[2], combined with the increasing number of machine learning algorithms in programs like Max/MSP[3], Pure Data[4], Chuck[5], EyesWeb[6] and the Wekinator[12]. As authors such as Fiebrink et. al.[12] have shown, it is now possible for performers to train a machine learning algorithm in real-time, live on stage and have the performer's movements (being sensed from anything such as a common gamepad to body worn accelerometers or EMG) be mapped directly to, for example, the synthesis parameters of a FM synthesiser. The machine learning algorithms featured in the programs listed above are generally excellent for solving two common problems; namely the discrete classification of a static posture as one of K possible postures and the mapping of an input signal to one or more continuous output variables, also known as regression. However, many of the existing machine learning toolboxes are still unable to classify patterns that occur in a multidimensional space and change over a variable time period, otherwise known in the machine learning literature as multivariate temporal signals. Further, it was found that the few toolboxes that do include multivariate temporal recognition algorithms either only work offline, take an extensive amount of time to train or are limited to accept a specific form of sensor input, such as the 2D data from a mouse or 3D data from an expensive motion capture device.

This has therefore provided the motivation for the design and development of a novel machine learning toolbox that can recognise static postures, perform regression and classify multivariate temporal gestures. The toolbox, called the SARC EyesWeb Catalog (**SEC**), has been specifically designed to work with any-type of N-dimensional signal and operates as a middleware application; thus enabling a performer to easily integrate it into their own existing composition and performance environment. The SEC has been designed so it is suitable for both performers with basic technology skills and no knowledge of machine learning right through to domain experts who want to create their own custom-built recognition systems. It is also suitable for researchers who wish to integrate and test a new form of feature extraction algorithm with the existing machine learning algorithms in the SEC. One of the main benefits that the SEC offers a performer is that it enables them to use the raw data or features from any sensor to quickly train a machine learning algorithm with the gestures the performer wants to use. After training the machine learning algorithm the performer can then use it to recognise their gestures in real-time, even in a continuous stream of data that also contains non-gestural data. In this paper we present the SEC and describe how its machine learning algorithms have been specifically adapted for the recognition of musical gestures.

2. RELATED WORK

Machine learning algorithms have been successfully applied to a number of tasks throughout many areas of musician-computer interaction (**MCI**). Lee et al. [17] and Fels et al. [9] were some of the first to apply the broad history of machine learning research on Artificial Neural Networks (**ANN**) to the field of MCI. Lee used an ANN to map the input from a radio baton, sensor glove or a MIDI keyboard to audio output and Fels mapped the input from a Cyberglove, 3-D tracker and a footpedal to a speech synthesiser. Fels work was later extended by Pritchard [22] who used several ANN to allow the user to synthesise audio, speech

[1] http://uk.wii.com/
[2] http://www.dcs.gla.ac.uk/research/shake/
[3] http://cycling74.com/products/maxmspjitter/
[4] http://puredata.info/
[5] http://chuck.cs.princeton.edu/
[6] http://www.infomus.org/EywMain.html

NIME'11, 30 May–1 June 2011, Oslo, Norway.

and song in real-time. Modler [19] also applied an ANN to map the sensor data captured by a sensor glove to continuously control the parameters of a synthesis engine running in SuperCollider. Along with applying the ANN to continually map the glove data to synthesis parameters, Modler also used the ANN to recognise patterns in the glove data, such as the classification of certain hand postures like thumbs up or an extended index finger. The recognition of a specific symbolic hand gesture could then be used to trigger a sound, with the energy of the finger movement being mapped to control the damping factor of a plate model. Cont et al. [6] created a number of ANN blocks for the Graphical User Interface (**GUI**) program Pure Data that enabled a performer to quickly train and recognise dynamic temporal gestures sensed by two perpendicular accelerometers. The network was trained using six constant speed circle gestures and was able to satisfactorily recognise a large variety of circles performed at different speeds and sizes.

Merrill et al. [18] built the FlexiGesture, a two handed device that features a number of sensors including 3-degree-of-freedom (**DOF**) accelerometers, 3-DOF gyroscopes, 4-DOF squeezing, 2-DOF bending and 1-DOF twisting. The user could train the system to recognise a temporal gesture by pressing a 'trigger' button which starts the data recording process, releasing the button when they have completed the gesture. The system then asks the user to continually reperform the gesture as it trains a template model for that gesture. Dynamic Time Warping was used as the recognition algorithm and tests showed that the system was able to classify novel gestures into one of ten classes with up to 98% accuracy.

Fiebrink et al. [12] created a real-time, on-the-fly machine learning-based system called the Wekinator that can be trained by the user in a number of seconds. The Wekinator affords the user the ability to quickly experiment with input/output mappings and even form judgements on the quality of the mapping by training and running it in real-time and observing the sonic results. The system was used for a live performance in which six performers started the training/mapping process from scratch, live on stage, and each performer gradually converged on the mapping setup they wanted as the piece progressed. Fiebrink et al. [11] extended this work by adding an additional 'play-along' paradigm to the Wekinator in which the user listened to a specific piece of music whilst mimicking the gesture they would have liked to have performed to make that sound. The system was then trained on this gesture-sound relationship and the user was able to create a sound or effect by performing the corresponding gesture.

Bevilacqua et al. [1] [3] have developed a real-time continuous gesture recognition system for Max/MSP in which a Hidden Markov Model can continuously output, not only the likelihood of the user performing a given gesture at the current time, but also, where in that gesture the user might be. One of the main benefits of this system is that it has been specifically designed to be trained with the minimum possible training examples (in some cases even one example can be sufficient). Bevilacqua et al. also [2] developed the MnM toolbox for Max/MSP which is dedicated to mapping between gesture and sound, applying algorithms such as Principal Component Analysis to reduce the dimensionality of the data, thus simplifying the mapping procedure.

A number of researchers have focused on capturing the natural gestures performed on acoustic instruments such as Overholt et. al. [21] who added a number of algorithms from the OpenCV library to their Multimodal Music Stand System (MMSS) to recognise the gestures of a flautist and use these to control a Max/MSP patch. Morales-Mazanares et al. [20] also tried to recognise the gestures of a flautist, using a probabilistic model to estimate what the attacks or angular displacement of the instrument could inferrer about the player's gestures. The accurate classification of violin bowing gestures has also received attention from [23] [26] [10]. Finally the recognition of a conductors gestures has received a large body or research [24] [16] [7].

These examples have illustrated how machine learning algorithms have been successfully applied to solve both classification and regression problems throughout many areas of MCI. A large majority of this work, however, has been designed for custom-built hardware devices [9] [18] or is constrained to recognising gestures from a specific sensor, such as the data from a 2D mouse [3], or is designed for a particular instrument, such as a flute [21]. This provided the motivation for us to develop a new machine learning toolbox that is specifically aimed for the real-time recognition of musical gestures. The SEC contributes to this existing work because it has not be constrained to work with just one sensor device or audio environment, can be used to classify both static postures, temporal gestures and perform regression and most importantly can be quickly trained by a musician with a small number of training examples.

3. THE SEC

The SEC[7] has been fully integrated as a third party library within a free program called EyesWeb. EyesWeb is an open software platform that was established to support the development of real-time multimodal distributed interactive applications and already features a large number of algorithms for processing both video and audio signals [5]. EyesWeb is a GUI orientated program that runs in Windows[8] which features a *patch window* onto which the user can drag a number of *blocks* that represent a specific algorithm or function. A block will commonly feature a number of input, output and parameter pins, with one block's output pin being connected to another block's input pin to create a signal flow between the two respective blocks. Using a small number of blocks in EyesWeb, for example, a performer could build a patch to capture real-time data from a sensor unit, filter the data and plot the results without having to write a single line of code. EyesWeb also enables any performer with more technical skills to develop their own blocks, which may be required to perform a specific type of feature extraction or to interface with a custom-built piece of hardware. All the blocks in EyesWeb are written in C++, giving the developer the ability to write fast, efficient code which is a necessity for real-time machine learning due to the large number of calculations required. EyesWeb therefore provides an excellent environment for both technical and non-technical users as complex signal processing operations can be easily constructed by connecting a number of blocks together or, alternatively, a custom block can be developed to perform one specific task.

3.1 The SEC Blocks

The SEC contains over 80 blocks (almost twice the number of blocks since its first public release [13]), all of which have been specifically designed for the real-time recognition of musical gestures. Along with featuring a number of rudimentary blocks for saving/loading data, converting from one data type to another etc., the SEC also contains blocks for signal processing, performing mathematical operations, and interfacing directly with hardware sensor units such as the Wii, the SHAKE and Infusion System's Wi-

[7] `http://www.somasa.qub.ac.uk/~ngillian/SEC.html`
[8] EyesWeb is currently being ported to Linux

Table 1: SEC Gesture Recognition Algorithms

Algorithm Name	Suitable Application	Learning Type
Adaptive Naïve Bayes Classifier	Classification	Supervised
Artificial Neural Networks	Regression	Supervised
Hidden Markov Models	Classification	Supervised
N-Dimensional Dynamic Time Warping	Classification	Supervised
Fuzzy C-Means Clustering	Classification	Unsupervised
K-Means Clustering	Classification	Unsupervised
K-Nearest Neighbor Classification	Classification	Supervised
Support Vector Machines	Classification	Supervised

microDig[9]. The SEC contains a large number of machine learning algorithms that can be used to classify static and temporal gestures as well as perform regression, a list of the main algorithms can be found in table 1. The SEC also contains a number of pre-processing or feature extraction algorithms that can be applied to reduce the computational load and complexity of a recognition problem along with a variety of post-processing algorithms that can be used to improve the overall system's classification performance. Each machine learning, feature extraction and post-processing algorithm in the SEC has been encapsulated as a single block, enabling the user to quickly create their own recognition system by dragging the algorithms they wish to use onto the EyesWeb patch window and connecting them together. This facilitates a user with even basic technical skills to apply a wide range of extremely powerful machine learning algorithms, such as Support Vector Machines (**SVM**) or Hidden Markov Models (**HMM**), to recognise their musical gestures without having to write a single line of code. One of the most important features of the algorithms within the SEC is that they have all been designed to work with any N-dimensional input signal. This means that the recognition algorithms are not constrained to just work with the two-dimensional data from a mouse for example, but can work with any N-dimensional continuous stream of data. This is a key advantage for performers, particularly those that create their own custom built sensors, interfaces or instruments, as the output from any sensor(s), or features derived from this sensor data, can easily be used as input to any of the SEC blocks.

3.2 Using the SEC for MCI

The machine learning algorithms within the SEC enable any performer to use one or more musical gestures to control and manipulate the performer's composition or improvisation software in real-time. For example, a musician could use a classification algorithm such as N-Dimensional Dynamic Time Warping (ND-DTW) [15] to classify a specific conducting gesture and use the recognition of this movement to trigger the computer to start manipulating the live audio recording of the musician the gesture was directed towards. At the same time, the performer could use a regression algorithm like an ANN to continuously map the velocity at which the performer made the conducting gesture to control the degree of the warping effect on the live audio recording.

Rather than targeting the SEC for just one specific piece of audio software, it has been designed to function as middleware enabling the user to pipe their sensor data into the SEC via a number of standard communication protocols, such as Open Sound Control (**OSC**) [25]. After recognition the classification results can be piped out of the recognition system to control any piece of audio or visualization software that use the same communication protocols. A middleware design architecture also enables the SEC to run on an independent machine from that which is running the audio software; which is beneficially for CPU intensive recognition algorithms. A performer can therefore write their own software to capture and parse the real-time data from whatever sensor(s) they might be using and pipe this data into EyesWeb via OSC. Alternatively, a performer could directly implement the sensor interface as an additional EyesWeb block.

3.2.1 Creating a Robust Recognition System

Machine learning algorithms rarely exist in a vacuum [8]. A robust recognition system commonly requires an appropriate pre-processing or feature extraction stage prior to any classification by a trained machine learning algorithm, with the predicted classification label being post-processed prior to being acted upon. A user may therefore want to experiment with various feature extraction algorithms or post-processing functions as well as testing which machine learning algorithm works best for the recognition of their gestures. It is for this reason that each feature extraction algorithm or machine learning algorithm has been encapsulated as a single EyesWeb block as this enables the user to connect the blocks together to create the recognition system the user thinks maybe most appropriate for solving their recognition problem. One of the major advantages of using a patch-based GUI program such as EyesWeb is that multiple recognition algorithms can be used in parallel, with the output of one classifier providing contextual information for another classification chain. For example, the predicted event of one classifier could be used to permit/deny the output of a second classifier from being acted upon.

3.2.2 Training a Machine Learning Algorithm

Prior to using any machine learning algorithm it must first be trained. This can be achieved by using a number of examples, called a *training set*, to tune the parameters of the algorithm's adaptive model or function. The training set could contain, for example, a number of recordings of each of the G gestures the performer wants the algorithm to recognise. Each training example may also be hand-labelled by the user in which case the problem is known as *supervised learning*. By adopting a machine learning approach, a musician can teach a computer to recognise their musical gestures by performing a number of repetitions of each gesture and use this data to train a machine learning algorithm. If the appropriate feature(s) are used to represent the gesture and a suitable algorithm is trained then the algorithm should be able to classify a new input vector as one of the G gestures it was trained with; even if the new input vector was not contained in the original training set. The ability to categorize correctly new examples that differ from those used for training is known as *generalisation*. In

[9]http://infusionsystems.com

practical applications, the variability of the input vectors will be such that the training data can comprise only a tiny fraction of all possible input vectors, and so generalisation is a central goal in pattern recognition [4].

The SEC features a number of useful tools to facilitate a user to efficiently create a training set and then quickly train a machine learning algorithm. Each algorithm, for example, will commonly have a dedicated block for recording training data, a second block for training the algorithm and a third block for the real-time classification of any new data using the trained model. This three block design enables the user to create a 'training patch' for recording and training the algorithm and a separate 'prediction patch' for real-time classification that may also contain other trained machine learning algorithms, post-processing algorithms and network connections to communicate with other audio/visual software.

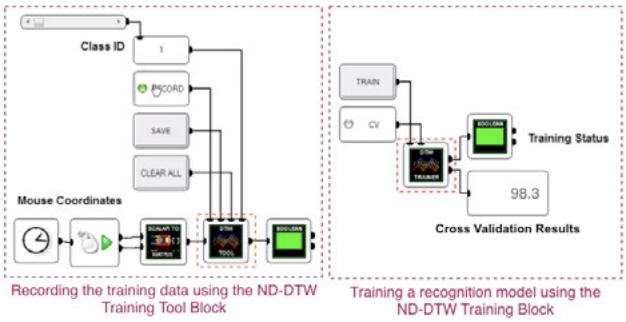


Figure 1: An example training patch for the ND-DTW algorithm

The training patch, illustrated in Figure 1, will have the identical sensor input and feature extraction methods as the prediction patch, see Figure 2, but can also contain a number of helpful features that assist the user in collecting and labeling the training data. This could consist of timer functions, for example, that enable the user to press a key to prepare the system to record a two-handed gesture. After a predetermined delay the user can then start to perform the gesture while the system records the training data, automatically labeling each training sample with the ID value of that gesture. After a further predetermined delay the system stops recording the gesture and the user can either record another example of the same gesture or move onto the next gesture in their vocabulary. When the user has created a number of training examples for each gesture they can save the training data to a file and then use this to train the machine learning algorithm. Each algorithm will then save its trained model to a file to enable it to be loaded by the real-time classification block.

The user can select if they wish to train the algorithm using an automatic validation method, such as K-fold cross-validation, to estimate the generalisation ability of the trained model or if they want to devote all of the available data to training the model and instead test the algorithm 'online' using the prediction patch. Either way, if a poor model has been created the user can quickly reload the original training data in the training patch and modify some of the parameters of the machine learning algorithm or even change the feature extraction method and quickly retrain a new model with the updated settings. Alternatively, the performer could use the one training set to train and validate several algorithms each with different settings all at the same time to determine the best features/algorithm/settings to use. The performer then simply needs to load the best model into the predication patch. These examples illustrate the advantages of using three separate blocks to create a training set, actually train a model and finally perform real-time prediction on new data using the trained model.

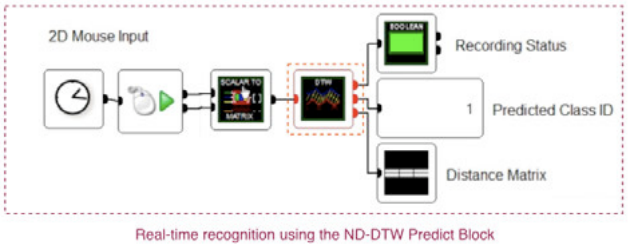


Figure 2: An example prediction patch for the ND-DTW algorithm

4. DESIGNING ALGORITHMS FOR MCI

The design, development and evaluation strategies applied to the algorithms within the SEC have required a fundamental paradigm shift from the common strategies employed throughout other areas of machine learning. In many areas of machine learning and gesture recognition the primary goal of any algorithm is to achieve a low inter-personal generalisation error, which is the algorithms ability to correctly classify the gestures of a new participant that was not included in the training set. This is the case, for example, with a computer game that recognises the gestures of a player and uses these to control a character in the game. In order for the system to robustly recognise thousands of different players across the world who may want to play the game, the recognition algorithm may need to be trained with a very large number of training examples collected from perhaps hundreds of different participants. The machine learning algorithm can then be trained with this extensive data set, perhaps overnight, after which it could be tested with another large test set that contains new data that was not used to train the algorithm to validate the classification abilities of the model.

4.1 The Intra-personal Generalisation Goal

Applying this approach to training and testing a machine learning algorithm may not be suitable, however, for the recognition of musical gestures - such as in a NIME interaction scenario. This is because each performer may want to define their own unique gestural vocabulary, i.e. the relationship between a gesture and its corresponding action. A performer may also want to capture the gestures using their own specific sensor device and use the recognition of a gesture to control a custom-made piece of audio software. It is therefore difficult to create the pre-trained recognition systems that are common throughout many areas of human-computer interaction (HCI). Musician-computer interaction alternatively requires a system that has a flexible input/output configuration and that can be trained by the performer using gestures from the their own vocabulary.

A user-configurable recognition system for MCI would not therefore require the inter-personal generalisation abilities found in other areas of HCI; instead it would simply need to provide a good intra-personal generalisation for the one performer that initially trained the system. If another performer wants to use their own input device or gestural vocabulary to control the same audio software, then they simply have to retrain the machine learning system with their own gestures. This intra-personal generalisation goal, which is quite a paradigm shift from many areas of machine learning and HCI, would not only offer the performer the advantage of being able to use their own hardware to capture gestures from their own gestural vocabulary and use

these to control their own specific audio software, it would also result in the requirement for a lower number of training examples per gesture - leading to a reduced amount of time spent in data collection and computational-training time.

4.2 Rapid Training, Testing & Prototyping

Any machine learning algorithm that can be quickly trained with a small number of training examples is extremely beneficial to a performer. An efficient training phase enables a performer to quickly decide upon a possible gestural vocabulary to use, train the recognition system and then, importantly, test the real-time prediction abilities of the system by performing the gestures and checking if they are correctly classified. Testing the system in this manner not only validates if a robust intra-personal generalisation error has been achieved, it also tests the aesthetic and practical validity of the gestures themselves. If a performer is unhappy with either then they can either change the feature extraction method or parameters of the machine learning algorithm being used and retrain the model. Alternatively the performer could scrap one or more of the gestures and replace them with more suitable movements. A recognition system that can be quickly trained and tested allows a musician to rapidly prototype any action-sound relationship they think may be useful for a real-time performance scenario, test the validity of such gestures and then focus their time on the musical elements of the performance instead of spending hours training a system to recognise their gestures only to find that the gestures do not work aesthetically or practical.

4.3 Validating An Intra-Personal Classification Algorithm

An intra-personal generalisation error would call for a new method of evaluating the classification abilities of a machine learning algorithm for MCI. For example in most machine learning applications, a large amount of data is collected from perhaps hundreds of users and the data is split into a training set and a test set. The machine learning algorithm is then trained with the training set and evaluated with the test set. If the training data is difficult or expensive to acquire, then a hold-out validation method such as K-fold cross-validation is used instead. Both of these validation methods are suitable for estimating the generalisation abilities of an algorithm that will be used in an interpersonal recognition system. For MCI however, a more suitable generalisation metric would be to use the average cross-validation error (**ACVE**) calculated by independently computing the cross-validation error for each of the P participants and then averaging this result. The ACVE is suitable for MCI because it can accurately estimate the intra-personal generalisation abilities of a machine learning algorithm, while at the same time being validated by a large number of different users.

In addition to using a quantitative error function, such as the ACVE, qualitative subjective measures can also be particularly useful for validating an algorithms potential application for MCI. This is because, as highlighted in [10], cross-validation-based approaches may be problematic under certain circumstances, such as overestimating a models quality when there are errors in the training data. In addition, cross-validation does not capture user-specific and subjective notions of cost (e.g., whether the classifier makes a mistake on an input that is highly likely to occur in performance, or on an input that can be avoided). Therefore combining both quantitative and qualitative error measures provides an appropriate method for validating algorithms for MCI.

4.4 The SEC Design Goals

Creating a recognition system that has a flexible input/ouput configuration, can be easily trained with examples from the user's own gestural vocabulary and that achieves a low intra-personal generalisation goal have therefore been the key objectives in the design and development process of the SEC. These objectives not only informed the design of the SEC blocks but they also challenged us to adapt a number of existing machine learning algorithms and develop some novel recognition algorithms specifically for musician-computer interaction. The algorithms were adapted and developed because we wanted each algorithm to be able to:

- Classify any N-dimensional signal and not be constrained to just working with one type of sensor
- Be quickly trained with a small number of training examples for each gesture
- Be capable of recognising a gesture from within a continuous stream of real-time data that also contains non-gestural data without having to train a *null-class*, such as a noise or silence class that is used in speech recognition
- Classify both static and temporal musical gestures

Several existing machine learning algorithms were adapted to meet these criteria by developing specific feature extraction methods that enabled any N-dimensional signal to be quantized and used as input to the algorithm. The training time of the HMM algorithm, for example, was significantly improved by developing multi-threaded training routines so that a unique thread was created to train the model for each individual gesture. The HMM algorithm was also adapted so that a classification threshold was computed for each gesture in the model, thus enabling the algorithm to reject any null-gesture if the log-likelihood estimate for that gesture was below a given threshold.

The SEC also features a number of novel classification algorithms that have been developed specifically for MCI, such as the Adaptive Naïve Bayes Classifier (ANBC) [14] and N-Dimensional Dynamic Time Warping (ND-DTW) [15]. Both algorithms have been specifically designed to be quickly trained with a small number of training examples, with the average training times for both algorithms on a small sized vocabulary of 10 gestures of just a few seconds.

5. APPLICATIONS OF THE SEC

5.1 Real-Time Improvisation For Piano

The SEC is being used in a collaboration between the first author and the UK based pianist Sarah Nicolls[10] to facilitate an improvised piece that is built only from live sampled piano, controlled entirely by the pianist's gestures. During the piece, the recognition algorithms in the SEC enable the performer to use subtle hand gestures to 'save' an improvised theme to an area of space located at various points above the piano keys. After a theme has been 'saved' to a space, the performer can then revisit this space at anytime and perform a number of other fine-grain hand gestures to playback the theme, warping, stretching, looping and filtering the sample all via gestural control.

5.2 Gestural Diffusion

The SEC machine learning algorithms are currently being used in an electroacoustic composition by Robyn Farah for live gestural diffusion. For this piece, the SEC algorithms

[10] www.sarahnicolls.com

have been trained to recognise a number of gestures that enable the performer to 'throw' a sound into the sonic space, with the trajectory and intensity the of gesture being used to control a number of parameters of the behavior of the sound. The performer can also use a number of two handed 'sweeping gestures' to control a large body of sounds that have already been introduced to the sonic space, pushing and pulling them around the space or removing them all together.

5.3 RadioStreams

The SEC is being used as the recognition system for an interactive installation piece called *RadioStreams*. In RadioStreams, a user can navigate around a virtual globe using intuitive pointing gestures and listen to live radio streams from each country they navigate through. If the user likes a radio station they can 'grab' and 'throw' it onto one of eight speakers located around them. When all eight speakers have been populated with a radio station the user can then create a live improvisation by playing and controlling the live radio stations using gestures similar to that of a choral conductor.

6. CONCLUSIONS

This paper presented the SEC, a machine learning toolbox that has been specifically developed for musician-computer interaction. The SEC features a large number of machine learning algorithms that can be used in real-time to recognise static postures, perform regression and classify multivariate temporal gestures. We also provided the motivation for the algorithms used for the recognition of musical gestures to achieve a low intra-personal generalisation error, as opposed to the inter-personal generalisation error that is more common in other areas of human-computer interaction.

7. REFERENCES

[1] F. Bevilacqua, F. Guédy, N. Schnell, E. Fléty, and N. Leroy. Wireless sensor interface and gesture-follower for music pedagogy. In *NIME07*, pages 124–129, New York, NY, USA, 2007. ACM.

[2] F. Bevilacqua, R. Muller, and N. Schnell. Mnm: A max/msp mapping toolbox. In *NIME05, Vancouver, BC, Canada*, 2005.

[3] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. Continuous realtime gesture following and recognition. *Lecture Notes in Cimputer Science (LNCS), Gesture Embodied Communication and Human-Computer Interaction*, 2009.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Science and Business Media, Springer, 2006.

[5] A. Camurri, P. Coletta, G. Varni, and S. Ghisio. Developing multimodal interactive systems with eyesweb xmi. In *NIME07*, pages 305–308. ACM, 2007.

[6] A. Cont, T. Coduys, and C. Henry. Real-time gesture mapping in pd environment using neural networks. In *NIME04, Hamamatsu, Japan*, 2004.

[7] R. Dillon, G. Wong, and R. Ang. Virtual orchestra: An immersive computer game for fun and education. In *Proceedings of the 2006 international conference on Game research and development*, CyberGames '06, pages 215–218. Murdoch University, 2006.

[8] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Citeseer, 2001.

[9] S. Fels. Glove-talkii: A neural network interface which maps gestures to parallel formant speech synthesizer controls. *CHI'95*, 1995.

[10] R. Fiebrink. *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance*. PhD thesis, School of Computer Science, Princeton University, 2011.

[11] R. Fiebrink, P. R. Cook, and D. Trueman. Play-along mapping of musical controllers. *The International Computer Music Conference (ICMC)*, 2009.

[12] R. Fiebrink, D. Trueman, and P. R. Cook. A meta-instrument for interactive, on -the-fly machine learning. *NIME09*, 2009.

[13] N. Gillian, R. B. Knapp, and S. O'Modhrain. A pattern recognition toolbox for musician computer interaction. *NIME09*, 2009.

[14] N. Gillian, R. B. Knapp, and S. O'Modhrain. An adaptive classification algorithm for semiotic musical gestures. In *the 8th Sound and Music Computing Conference*, 2011.

[15] N. Gillian, R. B. Knapp, and S. O'Modhrain. Recognition of multivariate temporal musical gestures using n-dimensional dynamic time warping. *NIME11*, 2011.

[16] A. Hofer, A. Hadjakos, and M. Muhlhauser. Gyroscope-based conducting gesture recognition. *NIME09*, 2009.

[17] M. Lee, A. Freed, and D. Wessel. Neural networks for simultaneous classification and parameter estimation in musical instrument control. *Adaptive Learning Systems*, 1706:244–255, 1992.

[18] D. J. Merrill and J. A. Paradiso. Personalization, expressivity, and learnability of an implicit mapping strategy for physical interfaces. *Proceedings of CHI 2005 Conference on Human Factors in Computing Systems*, 2005.

[19] P. Modler, T. Myatt, and M. Saup. An experimental set of hand gestures for expressive control of musical parameters in realtime. In *NIME03*, 2003.

[20] R. Morales-Mazanares, E. F. Morales, and D. Wessel. Combining audio and gesture for a real-time improviser. *in International Computer Music Conference, (Barcelona, 2005), ICMA.*, 2005.

[21] D. Overholt, J. Thompson, L. Putnam, B. Bell, J. Kleban, B. Sturm, and J. Kuchera-Morin. A multimodal system for gesture recognition in interactive music performance. *Computer Music Journal*, 33(4):69–82, 2009.

[22] B. Pritchard and S. Fels. Grassp: Gesturally-realized audio, speech and song performance. *NIME06*, pages 272–271, 2006.

[23] N. Rasamimanana, E. Fléty, and F. Bevilacqua. Gesture analysis of violin bow strokes. *Gesture in Human-Computer Interaction and Simulation*, pages 145–155, 2006.

[24] A. Wilson and A. Bobick. Realtime online adaptive gesture recognition. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 1, pages 270 –275 vol.1, 2000.

[25] M. Wright and A. Freed. Open sound control: A new protocol for communicating with sound synthesizers. In *International Computer Music Conference*, pages 101–104, Thessaloniki, Hellas, 1997. International Computer Music Association.

[26] D. Young. Classification of common violin bowing techniques using gesture data from a playable measurement system. *NIME08*, pages 44–48, 2008.

Music and Technology in Death and the Powers

Elena Jessop
MIT Media Lab
E14-333A, 75 Amherst Street
Cambridge, MA 02139
ejessop@media.mit.edu

Peter A. Torpey
MIT Media Lab
E14-333A, 75 Amherst Street
Cambridge, MA 02139
patorpey@media.mit.edu

Benjamin Bloomberg
MIT Media Lab
E14-333A, 75 Amherst Street
Cambridge, MA 02139
benb@media.mit.edu

ABSTRACT

In composer Tod Machover's new opera *Death and the Powers*, the main character uploads his consciousness into an elaborate computer system to preserve his essence and agency after his corporeal death. Consequently, for much of the opera, the stage and the environment itself come alive as the main character. This creative need brings with it a host of technical challenges and opportunities. In order to satisfy the needs of this storyline, Machover's Opera of the Future group at the MIT Media Lab has developed a suite of new performance technologies, including robot characters, interactive performance capture systems, mapping systems for authoring interactive multimedia performances, new musical instruments, unique spatialized sound controls, and a unified control system for all these technological components. While developed for a particular theatrical production, many of the concepts and design procedures remain relevant to broader contexts including performance, robotics, and interaction design.

Keywords

opera, Death and the Powers, Tod Machover, gestural interfaces, Disembodied Performance, ambisonics

1. INTRODUCTION: DEATH AND THE POWERS

The new opera, *Death and the Powers* [2], by composer Tod Machover, brings numerous artistic and technological innovations to the stage. In this show, the main character is the rich, powerful inventor and businessman, Simon Powers. Simon finds that he is dying and thus seeks to extend his life, legacy, and ability to interact with the world by uploading his consciousness, memories, and essence into a computer system built into his house. Powers' transformation from human being into the pervasive "System" occurs at the end of the first scene in the opera. The other characters in the opera—Powers' third wife Evvy, his daughter Miranda, his research assistant Nicholas, and representatives from the world at large—must learn how to relate to Simon in his new form. They question whether he is still alive and still the same person, and finally decide whether they wish to come join him in "The System."

While this story and the theatrical production involve a significant amount of technology, we wanted the story to be the primary focus. The technology needed to be in service of the story, not the story in service of the technology. Additionally, the technology had to be considerate of the needs of live theater: it had to be flexible, be expressive, and facilitate creativity.

This opera was developed by Machover's Opera of the Future group at the MIT Media Lab in collaboration with experts from the worlds of theater and film, including theater and opera director Diane Paulus and production designer Alex McDowell. Machover and the Opera of the Future group (formerly Hyperinstruments) have extensive experience with creating large-scale musical performances that incorporate significant technological innovations, including *Valis* [10], *Brain Opera* [15], and *Toy Symphony* [9]. The authors are students working with Machover at the Media Lab and were responsible for significant technical contributions to *Death and the Powers*.

The premiere performances of *Death and the Powers* were in Monte Carlo, Monaco in September 2010, with additional performances in Boston in March 2011 and Chicago in April 2011.

2. CONTEXT: TECHNOLOGY IN THE OPERA

As computer-based technology is such a a significant part of daily experience, it is now not unusual to introduce cutting-edge technology into performance. In fact, theater and performance artists have often been early adaptors of technologies from electric lighting to the Internet to digital video [5]. Technology has also found a place in the relatively new performance form of opera. While music, dance, and theater have been practiced for millennia, opera has its roots in 16th Century Italy. In fact, opera can be seen as a fairly new model of performance, still developing and still free for experimentation and exploration. Opera is also conducive to the integration of new technologies due to its history of incorporating elements from a variety of other performance traditions, combining musical performances, narrative storylines, theatrical design elements such as costume and scenic design, and occasional dances. Thus, a variety of opera productions and new operas have also incorporated technological performance elements into the medium. For example, Tod Machover's *Valis* [10] used two early hyperinstruments to create the musical score, with computer-generated music extending the live performance of a digital piano and a percussion instrument. *Lost Highway*, an opera based on the film of the same name by David Lynch, uses intricate live and prerecorded video streams and a rich synthesized soundscape to translate a complex movie into a compelling live musical performance. This production was directed by Diane Paulus with video design by Philip Bussman [6]. *StarChild* (1996) is an example of a "multime-

NIME'11, 30 May–1 June 2011, Oslo, Norway.

Figure 1: A rendering of the stage environment for *Death and the Powers*.

dia opera," incorporating surround-sound technology, planetary data sonification, and precise synchronization between a number of audio and video streams [12]. The Canadian director Robert Lepage has also brought interactive performance technologies into the world of opera, with works including his 2008 staging of Hector Berlioz's *La Damnation de Faust* for the Metropolitan Opera. This production used microphones to capture the pitch and amplitude of the performers' voices and the orchestra's music, as well as infrared lights and cameras to capture motion. The data from these sensors was used to shape projected images in real time, such as projected curtains waving behind dancers or giant projected flames that varied based on the singer's voice [14]. In contrast to many productions where single inputs are tied directly to independent outputs, we take a more abstracted approach to deriving expressive performance output from multiple live input streams. In examining the technology designed for *Death and the Powers*, it is also important to remember that most "high tech" theatrical performances simply use projection on screens and perhaps live camera feeds. If onstage performers' actions are measured, as in [4] and [14], that data typically is used to shape sound or visuals that share the stage with the measured performer. Additionally, theatrical technologies usually consist of discrete, disconnected systems. *Death and the Powers* features a distributed control system, an offstage performance translated into an expressive onstage presence, and a chorus of robotic characters. Through these elements and others, *Powers* extends the range of existing theatrical and operatic performance.

3. DISEMBODIED PERFORMANCE

One of the most unique theatrical challenges presented by the storyline of *Death and the Powers* is that, for the majority of the 90-minute opera, the main character is not represented by a physical actor onstage, but by the theatrical set. Powers' primary manifestation within the set takes the form of three fifteen-foot tall wall structures, or periaktoi. The structures can rotate and move freely about the stage. The walls represent book shelves and each book spine forms an LED display surface. The character of Simon Powers is expressed through a visual language created for this display surface, a language which develops and grows as Powers becomes more at home in The System.

A core performance issue is how to transform the character of Simon Powers from the physical form of our lead performer (James Maddalena in the premiere performances) into the theatrical environment. The stage must breathe, react, be emotionally expressive, and be as compelling as a human performer. One could have pre-recorded the singer's voice and have the behavior of the set and visuals on the stage be pre-scripted and triggered for separate scenes; however, we felt that it this would be constraining to the other performers and the orchestra and not expressive or conducive to the story or the performance. We determined it was a theatrical necessity to keep the power and presence of the singer's live performance, even though he would not be physically on the stage. Therefore, in our approach, the behavior of the scenic elements, including lighting, visuals, and robotics, are influenced in real time by the singer's performance.

Through a technique that we call Disembodied Performance, the singer's gestures, breath, and voice are observed and used to shape the output media on the stage in expressive and active ways. The Disembodied Performance System (DPS) consists of four separate layers: performance capture sensors (including both on-the-body sensors and audio sensors); data analysis software that transforms the raw data from performance capture system into meaningful abstractions; a mapping layer that relates the abstracted input parameters to parameters for output control; and an output layer, including visual, audio, and robotic elements, that shapes its behavior based on the control parameters. This system addresses a variety of questions about how to map a performance from one expressive modality, the human body, to a variety of other modalities, including non-anthropomorphic visual representations, lighting, movement, and sound [13].

3.1 Performance Capture and Analysis

3.1.1 Wearable Sensors

In order to measure the vitality and expressivity of a singer's physical performance, it was necessary to thoughtfully choose a set of performance capture sensors that would allow us to collect important features of the performance while not restricting the performer. We found that one of the key aspects of this physical presence is the performer's breath. The breath delivers information about musical phrasing, emotion, and a sense of life that would be evident to audiences watching the performer live on stage. Therefore, part of the sensor system includes a flexible band around the performer's chest that detects his inhalations and exhalations. The fabric band contains a stretch sensor located in a region of elastic fabric at the performer's back. As the performer inhales, his chest expands and therefore stretches the elastic region and the sensor. This simple sensor was found to detect information about the breath of the performer and his vocal phrasing that was more detailed than the information obtainable from audio or the score.

Accelerometers on the arms and the backs of the hands

are used to obtain information about the performer's gestures as he sings. Importantly, drawing on our group's background in Hyperinstrument design, we wanted to allow the singer to perform as he normally would onstage, with his training in how to use his body to convey a character's emotions. We thus chose not to capture specific gestures; more important was the overall character and expressive quality of his natural motion while singing expressively. We thus process the movement data into a set of higher-level parameters drawing on features of accelerometer data related to the quality of the movement (sharply changing, smooth, sudden, etc.). Such parameters are related to some of Rudolf Laban's qualities of movement, as discussed in the next subsection. All wearable sensors collect data with Funnel I/O microcontrollers and send that data wirelessly using the XBee protocol to external computers for analysis.

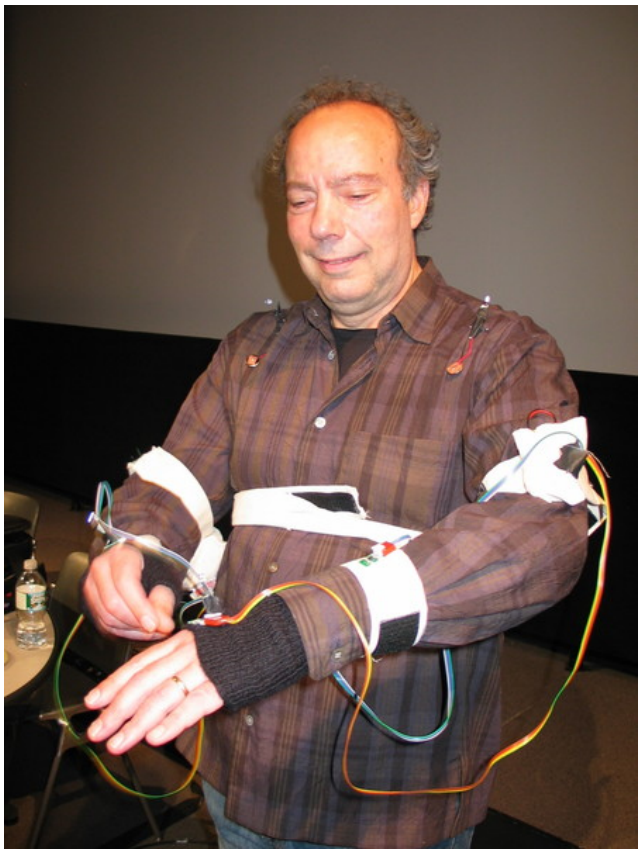


Figure 2: James Maddalena (Simon Powers) wearing prototype Disembodied Performance sensors.

3.1.2 Laban Effort Notation

In our analysis of movement data, we decided that the performer should not be required to use any particular movement vocabulary (essentially, his performance should not be choreographed); instead, the system should adapt to the performer's personal style of expression and augment his normal performance. We therefore chose to transform live movement data into information about the performer's emotionally-driven quality of movement.

We analyzed the qualities of movement using concepts borrowed Laban Effort Notion. Rudolf Laban held that the quality of any movement could be viewed as a point in a four-dimensional space, described by the four axes of Time, Weight, Space, and Flow [8]. The Time axis describes the speed at which a particular movement is being performed, from very fast and sudden to very slow and sustained. The Weight axis describes movement on a scale from firm to gentle. Firm movements are forceful, strong, resisting, heavy; gentle movements are relaxed, unresisting, light, weightless. Importantly for detecting this quality from sensor input, Weight is also a measurement of how much energy is being put into the movement. The third quality that Laban discusses is that of Space, which explores the way in which a movement travels through the space around the body, whether it moves directly or indirectly from one point to the next. Movement ranges on this axis from direct (moving in a straight line) to flexible (moving in curved, varying lines). The final quality of motion, Flow, is primarily descriptive of the amount of freedom of energy in a particular movement, reflecting how smoothly and continuously the movement is changing. This quality is on an axis from "fluid" movement to "bound" movement. In the Disembodied Performance System, immediate movement data and data trends are analyzed to locate the performer's movement in a three-dimensional quality space based on Laban's theories and defined by the axes of Time, Weight, and Flow, using techniques laid out in [7].

3.1.3 Vocal Processing

Additionally, vocal data from the performer was collected using microphones and used as input for audio processing. This vocal data, including both sung and spoken sounds, was analyzed for such audio parameters as amplitude, pitch, timbre, and purity of sound (consonance). These parameters, along with the parameters calculated from the breath data and movement quality analysis, are used as the inputs to the mapping system. In this way, the emotional content and quality of the actor's performance can be retained, but abstracted into a parameter space that is not tied to his physical body.

3.2 The DPS Mapping System

Interactive performances, where the live actions of a performer are captured by technology and used to shape visual, audio, or other aspects of a performance piece in real time, have a rich tradition in performance. In most interactive pieces, a pervasive and vitally important question is how the inputs from the live performance—sound, movement, location on stage, etc.—are mapped to parameters of the interactive output media. However, the systems that exist for creating these mappings are frequently limited by the small number of different mappings that can be created during the course of a particular piece, as well as by their focus on low-level sensor input rather than more meaningful abstractions of the input data. During our work on *Death and the Powers*, we developed a general-purpose mapping system to address these issues while additionally remaining sensitive to the needs of this particular piece and of the theater more broadly.

It was necessary for this mapping system to be very flexible and react appropriately to the fast-paced theatrical rehearsal process. The opera's systems are capable of creating an enormous variety of representations of Simon Powers; as those representations change from scene to scene and are developed during the course of rehearsals, the way that they are controlled by the live performance has to change as well. Additionally, the system needed to allow us to adjust the mappings between the live performance and the visuals immediately when given directions from the stage director Diane Paulus or visual notes from the production designer Alex McDowell, without having to stop the program or interactive output to make changes.

We developed a node-based flow interface that allows a user to create mappings by connecting streams of input

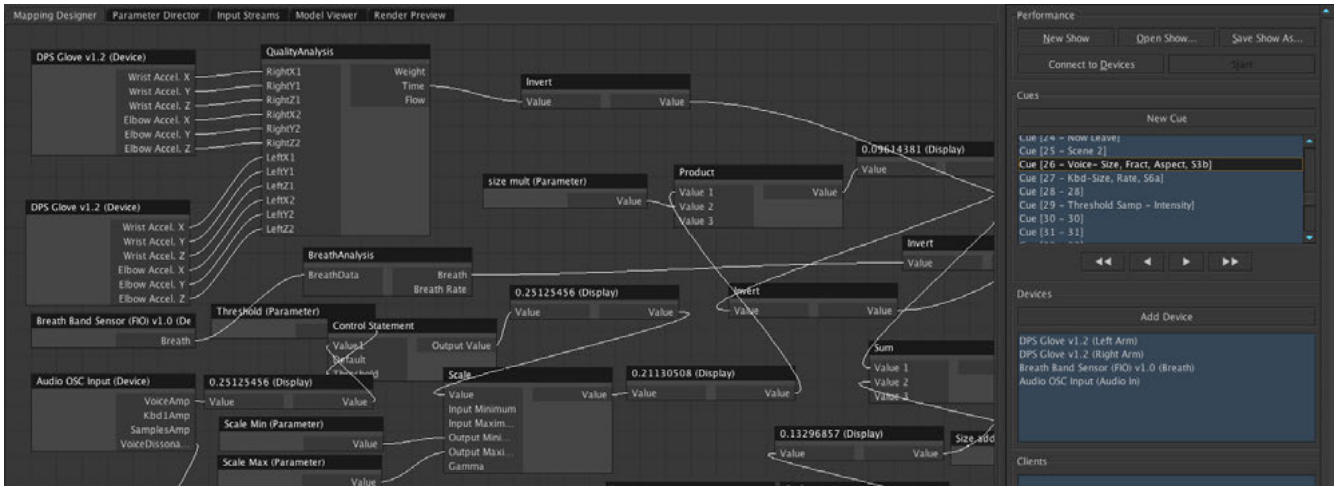


Figure 3: Mapping System screenshot.

data to outputs. Data analysis and arithmetic nodes allow for the creation of sophisticated transformations of performance data. All mappings can be manipulated and edited in real time. Each mapping between input and output data can be saved in a cue, allowing for mapping modes to be changed as needed during the performance. This mapping software, together with the performance capture sensors and a specially-designed visual display system, constitutes the Disembodied Performance System.

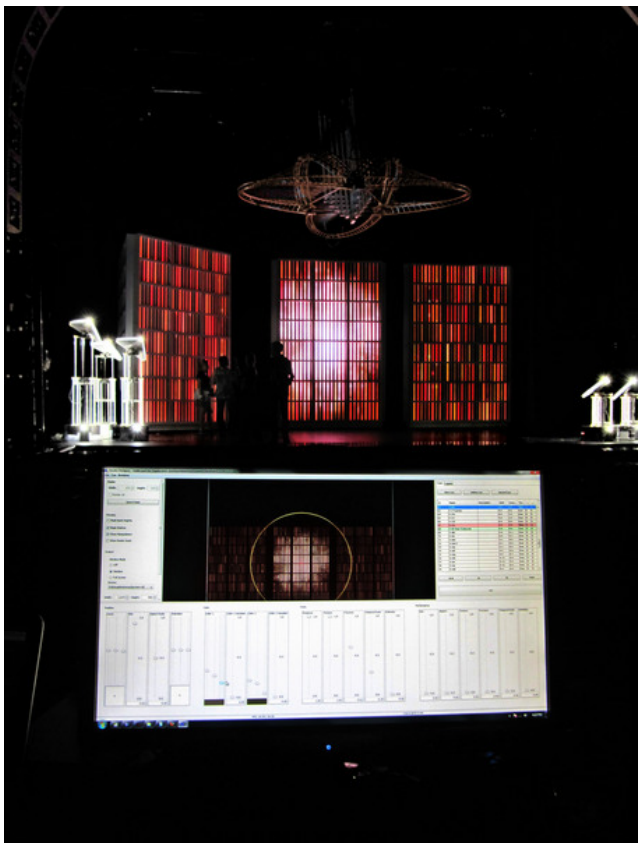


Figure 4: The Disembodied Performance System translates James Maddalena's performance into visuals on the walls.

3.3 The DPS Visualization System

As the primary representation of Simon Powers is through the visual displays on the three bookcase periaktoi, it was necessary to develop a system for creating visual content that could be not only shaped and developed with the production designer during the rehearsal process, but also modified in real time by live performance parameters. We accomplished this through the creation of a novel visual rendering system. This rendering software incorporates elements of a video compositing and animation environment, with the major organizing principle being the cue. Each cue can include compositions of procedural animation primitives, images, and pre-rendered video content. Additionally, triggering each cue puts the system into a particular mode where the live data from the performer procedurally shapes the generation of specific graphics and image. An operator can then mix the visual influence of the live performance on the preshaped cues using an Apple iPad during the performance. The result is a non-anthropomorphic manifestation of the actor's performance throughout the set.

4. OPERABOTS

Death and the Powers features two principal types of robots. The robots have multifaceted roles as set pieces, lighting elements, individual characters, and part of the manifestation of Simon Powers. The main periaktoi set pieces previously mentioned are three large robots. In addition to these robotic walls, the opera features nine smaller wireless Operabots, designed and fabricated at the MIT Media Lab. These robots, with triangular heads and bodies made of thin rods, can extend from four feet tall up to seven feet in height, have articulated heads, and use a holonomic omnidrive system that allows them to translate and rotate independently across the stage. Each Operabot has 11 expressive channels of LED lighting, including the bases, heads, and acrylic rods. The computational core of each of these robots is the small, efficient, and inexpensive One Laptop per Child XO computer.

In the context of the story, the Operabots serve as a Greek chorus, both characters in the action and commenting on it. In a prologue and epilogue, performed entirely by the robots, it is revealed that the robots habitually retell the story of Simon Powers, though they are still attempting to understand the idea of death. The Operabots also appear as figures throughout the main storyline.

As these robots play such major roles in the story, it was necessary to create a system for choreographing their

Figure 5: The nine members of the Chorus of Operabots.

behavior and movement that could allow the robots' performance to be shaped and developed along with the human performers during the rehearsal process. As Diane Paulus and choreographer Karole Armitage created choreography on robots, it was necessary for the robot system to quickly adapt to the demands of the moment and not limit the rehearsal process. The robots needed to be as flexible as the human performers. In order to accomplish this, we developed a new type of automation and control system specifically for theatrical robotics. The system combines timeline playback and parameter curves, familiar from typical animation software programs, with cuing, autonomous operation, procedural behaviors, and live control from a multitude of sources. This system has proven effective in choreographing the fast and complex Operabot movements and lighting, as well as the graceful repositioning of the three wall structures. Additionally, operators can assume control over any robot at any time, overriding any of its programmed behaviors. A 3D graphical simulation module in the choreographic software is included for assistance not only with monitoring the system during live control scenarios, but also to allow for offline programming and choreographing without needing to use the physical robots [11].

An ultra-wideband RFID absolute positioning system encompassing the stage tracks the location of operabots, walls, and singers, and communicates position information to each of the robots. This allows the robots to autonomously navigate along a predetermined trajectory and avoid each other and actors onstage, ensuring safe and robust operation. The nine Operabots may also be puppeteered as needed by operators situated above the stage, using commercially available video game controllers.

5. AUDIO SYSTEM

Death and the Powers is performed by a small ensemble with lightly amplified voices and accompanied by a 15-piece orchestra and electronic sound. To locate Simon Powers in The System, this production relies on sonic transformations in addition to visual transformations. To help express Powers' new omnipresence, his voice must be able to appear from anywhere, shifting location from moment to moment. The audio infrastructure to support this movement is quite extensive, utilizing two formats of surround sound, real-time performance control and several custom effects engines; all with the goal of achieving a smooth continuum ranging from acoustic to amplified textures. The large dynamic range in the audio system allows the most basic characteristics of the sound to follow Simon's emotions very closely.

5.1 Architecture

The heart of the *Death and the Powers* audio system is a digital signal processing (DSP) engine based on CoreAudio AudioUnits plugins running inside Digital Performer 7. This engine performs processing on the production's 350 audio inputs and 250 audio outputs. Custom plug-ins implement 3rd order ambisonic encoding and decoding and wave field synthesis (WFS) encoding and decoding. All DSP systems connect via three 64 channel bi-directional MADI fibre optic trunks to a Studer Vista 5SR mixing console, where the production is mixed on 12 VCAs. Each of the Duran Audio Axys loudspeakers in the front Left-Center-Right audio system runs additional DSP internally to manage crossovers, system-wide equalization and time alignment.

5.2 Localization Techniques

Death and the Powers uses two methods for localizing sound: ambisonics and WFS. The ambisonic system reproduces a consistent 3-dimensional sound-field over a large range of venues and speaker configurations. It is used to move voices and orchestral textures around the perimeter of the audience. The WFS system generates a wave-front where the origin of the wave is a location on stage. The WFS system is used with the tracking system to provide realistic reinforcement of the voices and robots. Natural amplification radiates from the location of the performers, instead of the WFS speaker array located along the front of the stage.

5.3 Seating Zones

Surround sound is an essential part of *Death and the Powers*, so it is critical for the entire audience to experience it. This presents a challenge in theaters where seating areas may be acoustically isolated (i.e. upper balcony or box seats). In this case, it is necessary to have a speaker system for each acoustic zone. In *Death and the Powers*, each zone contains a small surround sound system as well as supplementary front speakers. The system's DSP is tailored for the size and shape of the zone through the weighting of harmonic orders for the ambisonic decoders on each output. This maintains apparent resolution of each zone's surround system. For the premiere performance, 143 unique speaker outputs were used.

5.4 Control

All routing of sources to speakers is managed intelligently by the DSP engine and controlled live by streaming Open Sound Control (OSC) protocol [3] messages from Apple iPads and from the same RFID tracking system that tracks the robots. The DSP system utilizes a central shared memory where coordinates and audio are made available to all encoders and decoders. A network daemon listens for OSC and writes DSP parameters to the shared memory. An external system accepts tracking, cuing and remote control data, smooths it and defines which data is forwarded to the DSP network daemon.

6. THE CHANDELIER

Another of Simon Powers' manifestations in his environment is through the form of the Chandelier, a large stringed set piece that serves as a lighting element and, in a romantic scene in the middle of the opera, a musical instrument. For the first several scenes of the opera, the chandelier remains aloft and unmoving over the stage. As Simon's wife Evvy attempts to communicate with her husband in his new form, Simon inhabits the Chandelier and descends to wrap around Evvy. As she touches the Chandelier, it reveals itself to be a musical instrument. The Chandelier's primary sound is a complex electronic mix created from Simon's voice. When

Evvy strokes and strums the Teflon strings of the Chandelier, she controls this rich sound - bringing it out, dampening it, as if she's physically touching and manipulating his voice. Additionally, when she plucks the strings, she adds more processed string-like sounds to the mix. Stretch sensors wrapped around the tops of the Chandelier's strings detect the vibrations of the strings as they are played, allowing the performer in the role of Evvy to interact in a highly physical and sensual manner with the instrument.

Figure 6: Patricia Risley as Evvy with the Chandelier.

7. UNIFIED CONTROL ARCHITECTURE

All of the elements of the theatrical set—from the movement or robotic elements, spatialized sound, lighting, and visuals—must act in synchrony if they are to provide a consistent impression of a single expressive character. To accomplish this, the distributed show control systems are networked and interact by sharing data and interfacing with traditional theatrical controls. Data is exchanged over a common IP-based network infrastructure using OSC so that any system can respond to input from any other. Our protocol borrows from MIDI Show Control for cue-based and timeline navigation, as well as the Architecture for Control Networks (ACN) protocol [1]. Although robust ACN implementation was not readily available at the time we began creating the control systems for *Powers*, we did implement an ACN-inspired form of device description language and autodiscovery. Using OSC also meant that our novel systems could immediately exchange data with off-the-shelf audio software suites used in the production as well as user interfaces such as TouchOSC for the iPad. Additionally, the ranges of all data and control instructions are normalized into a range from -1.0 to 1.0 so the systems can logically communicate. With this system, gestures can be created across media: the sound of Powers' voice can match the movement of a visualization on the walls, or a robot's lighting can be driven by a performer's voice. The individual systems can be treated as parts of an artistic whole.

8. FUTURE DIRECTIONS

The technologies developed for and used in *Death and the Powers* have been designed so that they can be generalized for other performance contexts as well. The mapping system designed as part of the Disembodied Performance System can easily be adapted to allow the rapid development of mappings between any performance inputs and any control parameters for interactive systems. In fact, the Disembodied Performance mapping system has already been used in a piece for solo cello written by Tod Machover. In this work, "Spheres and Splinters", the sound of the cello and various properties of the cellist's bowing are used as inputs to the mapping system, which transforms those inputs into control of sonic transformations applied to the cello, and movement of sound in an ambisonic setup.

Additionally, many of the concepts and software systems developed for the opera are applicable for fields such as remote presence, storytelling, personal expression, and robotic control. For example, remote presence tools could be greatly enhanced the ability to capture emotional information about a person's movement and transform that to a set of expressive parameters for visual control. With tools such as those developed for *Death and the Powers*, subtleties and evocative details of physical movement can be used to create even richer interactions, performances, stories, and experiences: experiences that use and benefit from digital technology, but which are still inexorably linked to very human stories.

9. ACKNOWLEDGMENTS

Thanks to Tod Machover and the Opera of the Future Group, the MIT Media Lab, and the cast, crew, and creative team of *Death and the Powers.*

10. REFERENCES

[1] Architecture for control networks. http://www.engarts.com/acn/.

[2] Death and the powers website. http://powers.media.mit.edu.

[3] CNMAT. opensoundcontrol. http://opensoundcontrol.org.

[4] M. Coniglio. *New Visions in Performance*, chapter The Importance of Being Interactive, pages 5–12. Taylor and Franci, 2004.

[5] S. Dixon, editor. *Digital Performance: A History of New Media in Theater, Dance, Performance Art, and Installation*. MIT Press, Cambridge, MA, 2007.

[6] I. Hewitt. Lost highway: Into the dark heart of david lynch. http://www.telegraph.co.uk/culture/music/opera/3672082/lost-highway-into-the-dark-heart-of-david-lynch.html.

[7] E. Jessop. A gestural media framework: Tools for expressive gesture recognition and mapping in rehearsal and performance. Master's thesis, Massachusetts Institute of Technology, 2010.

[8] R. Laban. *Mastery of Movement*. Northcote House, 4th edition, 1980.

[9] T. Machover. Shaping minds musically. *BT Technology Journal*, 22(4):171–179.

[10] T. Machover. Hyperinstruments: A progress report, 1987-1991. Technical report, MIT Media Laboratory, 1992.

[11] M. Miller. Show design and control system for live theater. Master's thesis, Massachusetts Institute of Technology, 2010.

[12] J. Olivero and J. Pair. Design and implementation of a multimedia opera. *Proceedings of the 1996 International Computer Music Conference*, 1996.

[13] P. Torpey. Disembodied performance: Abstraction of representation in live theater. Master's thesis, Massachusetts Institute of Technology, 2009.

[14] D. Wakin. Techno-alchemy at the opera: Robert lepage brings his "faust" to the met. *New York Times*, November 2008.

[15] S. Wilkinson. Phantom of the brain opera. *Electronic Musician*, January 1997.

Design and Evaluation of a Hybrid Reality Performance

Victor Zappi
Istituto Italiano di Tecnologia
via Morego 30
Genoa, Italy
victor.zappi@iit.it

Dario Mazzanti
Istituto Italiano di Tecnologia
via Morego 30
Genoa, Italy
darmaz@gmail.com

Andrea Brogni
Istituto Italiano di Tecnologia
via Morego 30
Genoa, Italy
andrea.brogni@iit.it

Darwin Caldwell
Istituto Italiano di Tecnologia
via Morego 30
Genoa, Italy
darwin.caldwell@iit.it

ABSTRACT

In this paper we introduce a multimodal platform for Hybrid Reality live performances: by means of non-invasive Virtual Reality technology, we developed a system to present artists and interactive virtual objects in audio/visual choreographies on the same real stage. These choreographies could include spectators too, providing them with the possibility to directly modify the scene and its audio/visual features. We also introduce the first interactive performance staged with this technology, in which an electronic musician played live five tracks manipulating the 3D projected visuals. As questionnaires have been distributed after the show, in the last part of this work we discuss the analysis of collected data, underlining positive and negative aspects of the proposed experience.

This paper belongs together with a performance proposal called *Dissonance*, in which two performers exploit the platform to create a progressive soundtrack along with the exploration of an interactive virtual environment.

Keywords

Interactive Performance, Hybrid Choreographies, Virtual Reality, Music Control

1. INTRODUCTION

Influences from different disciplines strongly characterize contemporary art production, where theatre, dance, visual art and music often combine together to form novel artistic expressions. One of the resulting consequences of this wonderful process is the difficulty in making a neat distinction between interactive/real time performances and participatory installations; although previously separated, these two experiences merge, as the technical and conceptual arrangement of novel art pieces - the mise en scène - binds audience and performers with a powerful emotional stream.

More and more often technology is the basis of these changes, affecting the nature of the stage itself, blending paradigms, and extending the performance range with undiscovered expressive possibilities. Johannes Birringer defined the "digital dispositif" [2] as the comprehensive environment which conveys this extended notion of the stage, a platform where different media and data are captured and networked to define a dialogue between performers, audience and the "dispositif" itself. Birringer argues that "the imaginative range/freedom [of the performance] is to some extent driven or inspired by the arrangement that are made [in the digital dispositif]", but he adds that, at the same time, certain methodological restrictions or limitations may arise from the defined platform behavior.

The work presented throughout this paper stems from our interest in technology supporting art, especially concerning the way interactive multimodal setups could support innovative ways of expression, without interfering with the creative process. To this end we designed and developed a multimodal platform for Hybrid Reality live performances: exploiting 3D projection and motion capture technologies, artists and interactive virtual objects share the same real stage, creating choreographies where real and virtual world literally overlap. The created 3D environment embraces the spectators too, providing them with the possibility to directly modify the scene and its audio/visual features.

In Section 3 we discuss technical and conceptual details that define a Hybrid Reality performance, describing the guidelines we followed to transform our Virtual Reality (VR) room into a mixed-reality stage. In Section 4 we introduce *Virtual_Real*, the first audio/visual performance that took place in this complex environment; in this part the specific creative process is analyzed, exploring the technical and artistic solutions that characterized the performance as an interactive audio/visual concert. As questionnaires have been distributed after the show, in Section 5 collected data are presented, in order to perform an evaluation of the audience's experience, both from the perceptive and the emotional point of view.

2. RELATED WORKS

The primary characteristic of this project is the co-existence on the stage of a human element (i.e. one or more performers) and a machine element (i.e. the "dispositif"), which manifests itself through the interactive visual environment (Figure 1); both actors play in the scene, sharing the attention of the spectators in a duet which might enhance the expressive power of the piece. This concept has already been explored in impressive works. With *Glow* [4] the company Chunky Move presented a piece in which a dancer moved while lying on the ground, surrounded by a digital landscape generated in real-time in response to the performer's movement; the tracked body's gestures are extended by and in turn manipulate the video world that surrounds it, ren-

NIME'11, 30 May–1 June 2011, Oslo, Norway.

dering no two performances exactly the same. In Miwa Matreyek's *Glorious Visions* [12] the performer's shadow is projected onto a screen where an oneiric world comes to life, transforming her body into the center of gravity for small living creatures. Although in this case the "dispositif" is technologically rather simple, the result is highly absorbing and surreal. Low cost technologies networked with code-based frameworks are also exploited in the project *Euphorie* [1], a cross between a sonic installation and a musical gig; the performers play self-made instruments behind a transparent screen, where real time visuals were displayed.

Figure 1: The performer on stage, surrounded by the virtual environment: the lava planet follows his movements, always floating over his palm.

The second characteristic of our multimodal platform consists of the possibility to influence sound and music through the manipulation of the graphic environment, exploiting visuals as a new kind of musical instrument; this concept has inspired artists and researchers, and involved different experimentations on human-computer interaction technology. In *The Sound of One Hand* [8] Jaron Lanier performed live on a plain stage, wearing a head-mounted display and a single dataglove: immersed in a dramatic virtual environment he could play different kinds of virtual musical instruments, while his viewpoint was projected onto big screens for the benefit of the audience. Other wonderful examples are the *Iamascope* [5], by Fels et al., and the *Manual Input Sessions* performance [9], by Levin et al.: in these works hand gestures are tracked to mutate the projected graphic environment, and to dynamically create and control sounds.

The third characteristic of the platform is the active role the audience has during the performance. Since the beginning of the 60's art boundaries have broadened to embrace the participation of the audience; Allan Kaprow's *Happenings* [7] are the first examples of such an attitude, breaking down fixed structures and hierarchies that previously differentiated a performance from an installation. More recently technology and consumer electronics have been exploited to create participatory environments, as in the *Dialtones* [10] performance, again by Levin at al. In the works of Kaiser et al. [6] and Samberget al. [14] the possibility to improve the interaction between the VJ and the audience in dance clubs is investigated, thanks to a multimodal console accessible from the dance floor; while in *The Interactive Dance Club* [15] Ulyate et al. proposed a wonderful venue where 9 installations located all over the club permitted participants to influence music, lighting, and projected imagery, in zones for single participants, dual participants and groups.

3. DESIGN OF A HYBRID REALITY PERFORMANCE

3.1 Projections and Viewpoint

In order to achieve a consistent superimposition of virtual and real elements within the scene, we have made use of the technical setup available inside our department VR room, where a $4x2m^2$ Powerwall is in front of a $4x4m^2$ area. Here 12 IR cameras and an inertial ultrasonic tracking system can be exploited to track people and objects. Two 3D projectors synchronized with shutter glasses draw virtual objects, which appear to move off from the flat surface of the screen, invading the physical space in front of it.

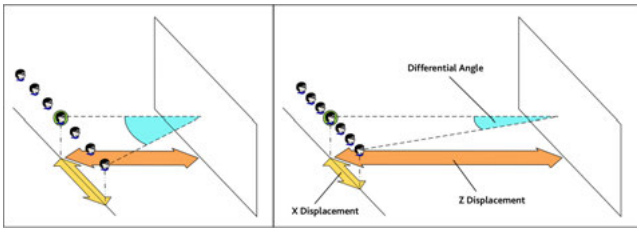


Figure 2: Increasing Z displacement (shown in dark orange in the schema) and diminishing X displacement (in light orange) we succeeded in reducing the differential angle (in blue) between the fixed center viewpoint and the actual spectators' viewpoints; a lower angle determines less visual distortion for side spectators.

This environment has been previously used only to perform single user experiments, to evaluate perception and interaction paradigms in virtual environments. Several observations of subjects interacting with virtual objects raised the idea to stage performances where interaction could be transformed into artistic expression, shown from the viewpoint of spectators that watch the artist while in the virtual environment. Thus we decided to discard from our system one of the essential features of projected VR, user's head tracking, together with the concept of user's viewpoint. We defined instead a fixed central viewpoint, shared among all the spectators and ideated to fit the position of a sitting person, who directly watches the screen from the audience's seats. Generally a shared viewpoint introduces an error in the correct perception of 3D objects, a distortion, especially during interactions; this error consists of a misalignment between the perceived vanishing points of the two superimposed scenes, the real one and the virtual one; its intensity is directly proportional to the X displacement between the fixed viewpoint and the audience viewpoint, while it considerably diminishes with increasing Z displacement, that is the distance from the virtual scene (Figure 2). Despite the extremely small space available in our VR room, we succeeded in creating an area where up to 9 spectators can comfortably take a seat and attend a performance with no noticeable visual distortions (Figure 3).

Thanks to this arrangement, in the eyes of the audience performers, real items and virtual object share the same physical space, on a stage where interaction discloses an infinite number of choreographic possibilities. According to Milgram's Taxonomy and Virtuality Continuum [13], we chose the term "Hybrid Reality" to define these performances, since real world and virtual world objects coexist, and "real physical objects in the user's environment play a role in (or interfere with) the computer generated scene". Nevertheless this definition doesn't completely fit the kind of arrangement we are presenting in this paper, in fact, as

Figure 3: Despite the small available space, the chosen arrangement permits to host up to nine spectators in our VR room.

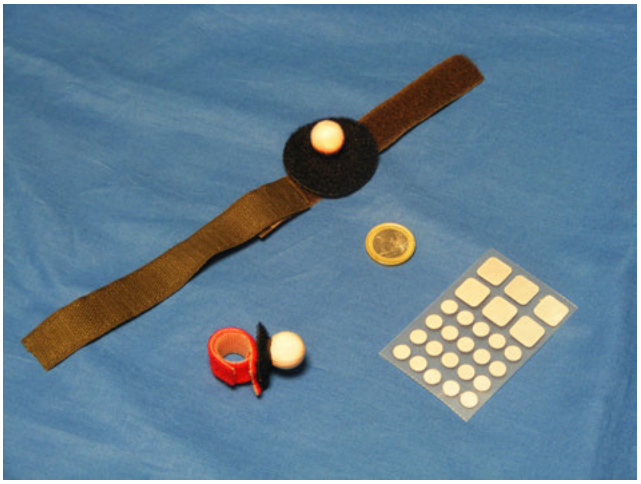


Figure 4: Small passive reflective markers tracked by the system: the adhesive version (on the right) does not need Velcro strap to be attached.

happened in similar performances like Kim Vincs and John McCormick's *Touching space* [16], the environment loses its egocentric connotation, becoming exocentric, separating the viewer from the user that actually interacts with virtual (and real) objects.

3.2 Hybrid Choreographies

As the overall arrangement loses its VR connotation, projected 3D images become a natural evolution of live stage visuals, not only accompanying the artist during the show, but embracing her/him. The complete arbitrariness in shape, position and behavior of these projected objects drastically enlarges the choreographic possibilities of Hybrid Reality performances, with respect to bi-dimensional visuals we are used to. Furthermore the brain of the "dispositif" that manages the virtual environment and all of its rules can be programmed, in order to lead all of these features into a meaningful relationship with the on-stage artists (i.e. Hybrid Choreographies).

Most of this work is developed in VRMedia[1] XVR, the central software on our platform. Primarily meant for VR application design, it proved to be a very flexible environment, thanks to a simple but powerful code syntax, and to the possibility to support custom cross-language modules (e.g. C++ dll's, Python scripts). External meshes, modeled with 3D graphic softwares like Autodesk[2] 3DStudio Max or Maya, can be imported into the virtual environment, including materials and animations; making use of GLSL scripts, these objects can be manipulated in real time, dynamically changing material properties and model geometry through fragment and vector shaders. Also physical behavior can be simulated, exploiting the Nvidia[3] PhysX module to give life to the environment, allowing the creation of worlds governed by real or unnatural physics laws.

XVR also processes and routes huge quantities of data coming and going from and to external hardware and software. Each device used on and off stage can be connected to this network, in order to synchronize it with the whole system, and to easily define its role within the performance. Such a client-server structure, easy to expand and to configure, has been already included in other live performances setups, like Last Man to Die's *Vital LMTD*[11].

[1]http://vrmedia.it/
[2]http://usa.autodesk.com/
[3]http://www.nvidia.com/

One of the most important sources of data within the platform is the low-latency IR tracking system, which broadcasts the positions of up to 50 passive reflective markers (Figure 4) moving within the stage area; thanks to the UDP connection between the built-in client and the main server, these data are stored and processed by XVR. These lightweight markers can be easily attached to the performer's body, in different configurations, to distinguish specific parts of the anatomy (e.g. hands, legs, head). In a 3D controllable environment, providing the system with information about artists in space, such as body pose or finger XYZ position, is fundamental to make virtual objects responsive to performer's movement, to make them communicate (directly or remotely) with other real and virtual subjects [3]: objects could move according to dancers' position in improvised choreographies, or they could be dragged directly by their hands, shattered or manipulated into new shapes (Figure 5). Furthermore tracking is not confined to humans, items that are physically located on the stage could carry markers and be utilized to trigger virtual interactions.

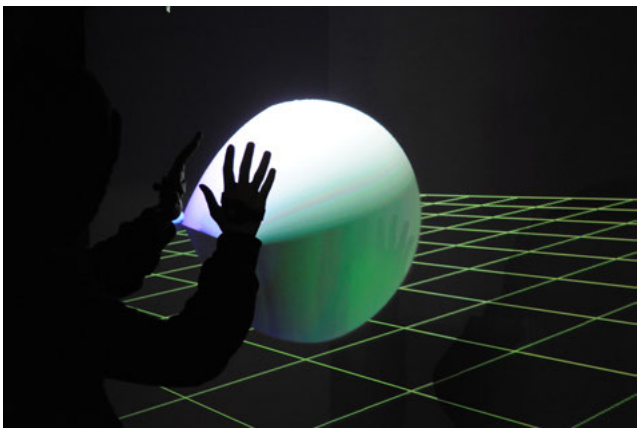


Figure 5: Through manipulation the performer can directly affect the shape of meshes, creating in real time unrealistic figures.

One of the most interesting features supported by our platform is the possibility to bi-univocally bind the visual and the sound environment: OSC and MIDI signals are sent and received through the network to exchange infor-

mation with softwares for audio synthesis and processing, external controllers and musical instruments. Sound is outputted onto a custom 14.1 audio system for sound spatialization, while external audio signals are acquired through a 26 in/out low latency audio interface. This scenario permits a real-time 3D visualization of music, programmable according to the preferred synaesthetic criteria. Moreover, in terms of interaction, it provides on and off-stage performers with the opportunity to manipulate sounds and music not only playing their musical instruments and controllers, but also physically interfering with the virtual environment; practically an infinite number of different metaphors can be created, using body motion capture as gestural input to control all the sound devices connected to the network, and providing visual and audio feedbacks both for performers and audience.

3.3 Audience's Participation

Tracking is also used to offer spectators an active role in the hybrid performance. To provide collaborative experience the audience's gesture recognition does not need to be as sensitive and precise as that of the performer; however a good resolution surely helps in distinguishing single spectator's different motions and intentions, allowing participation in a more engaging manner. Diverse solutions can be employed to achieve this goal according to the desired level of detail, even using two different systems on the same platform, one to monitor the stage, the other to monitor the seat area.

We successfully tested 2D silhouette extraction through a single RGB/IR camera and 3D volume reconstruction with a time-of-flight IR sensor, both working with no marker support needed. Because of the small dimensions of our VR room, we were also able to enlarge the detection area of the 12 IR camera tracking system used for the stage, extending marker detection up to the seat area; although heavily linked to the morphology of the place, this third solution proved to be convenient in terms of latency, resolution and technical ease, as no additional devices have been plugged into the system. Furthermore we strongly believe that a high sense of immersion - of inclusion - within the performance could be achieved permitting the single spectator to directly touch virtual objects, thus modifying the audio/visual environment. In a Hybrid Reality performance the presence of graphic elements is perceived as real, for they occupy a volume in a space that is real, and they support interaction with a real performer; so, as these objects travel through space getting closer, spectators look forward to reach them, to touch them, expecting to have interaction capabilities themselves. 3D tracking of spectators' hands allows the extension of interaction algorithms to audience's participation (including metaphors for sound creation as well), in order to create a collaborative multimodal domain, where the communal possibility to touch/modify the virtual environment works as a connection between the artists and the audience. This connection may be subjected to well defined rules, to highlight the artist and her/his starring role as opposed to spectators, or it may discard such a distinction, moving on the blurred line that divides performances from installations.

4. VIRTUAL_REAL

Virtual_Real has been the first Hybrid Reality performance designed and developed for our multimodal platform. Born from the collaboration with the electronic composer USELESS_IDEA[4], the performance stemmed from the artist's passion for both music and graphics as expressive means, which were combined together to transform a music concert into an experimental audio/video venue.

The on-stage setup was rather simple: in front of the screen we centered a table over which the performer installed his gear, consisting of a laptop, an USB MIDI controller and a small mixer, connected to the platform; the incoming audio signal produced by the musician was processed through Ableton[5] Live, extended with the LiveAPI/ LiveOSC package. The off-stage setup, although much more complex, was completely transparent to spectators, and included an Intersense[6] 3D wand, a Monome[7] (Figure 6), and the previously introduced multi-camera motion capture system. The performer also had a marker attached with a strip over his dominant hand; spectators were provided with a marker as well, mounted on a small ring to be put on top of their index finger.

Figure 6: A 3D wand and a 40h Monome assembled from a kit were plugged into the system as off-stage audio/visual controllers.

USELESS_IDEA played five original tracks, specifically composed for the event. Each track was associated to an immersive 3D choreography, arousing visual atmospheres directly connected to the sounds and the music. The artist actively participated in all the steps leading up to the final show, trying to explain his motivations and his messages, towards a keen refinement of algorithms, controls and contents. With the artist's agreement, we chose to alternate 3D visuals with short 2D sequences, in order to intensify the perceptive and emotional impact of virtual objects, as well as to gently blend in the eyes of the audience the classic paradigms of stage visuals with the unconventional immersive experience. Particular effort was put onto interaction design too: many algorithms were tested by the artist in order to define a set of simple but also powerful and visually impressive metaphors, to process sound manipulating the 3D visuals; consequently the musical pieces have been composed as modular structures, which encourage the building of live improvisation for visual interaction.

The result looked like a journey in five different scenarios, from deep space, to worlds of dancing and pulsing particle systems, where the artist could move objects as 3D XYZ faders, and trigger loops by touching and morphing unnatural shapes. Spectators were also engaged by this

[4]http://uselessidea.blogspot.com/

[5]http://www.ableton.com/

[6]http://www.intersense.com/

[7]http://monome.org/

journey (Figure 7): each object coming close enough to be reached supported interaction, as it could be moved, thrust aside, and sometimes manipulated in its visual characteristics (e.g. color, shape) according to the current scene rules. The related sonic manipulation has been limited to sound spatialization of some audio patterns, as the artist insisted on keeping the venue as similar as possible to a live concert, where music is exclusively played by the performer. Just behind the audience, an off-stage performer supported the artist, triggering scene changes and manually controlling some parameters of visual choreographies, utilizing the Monome and the wand plugged into the network.

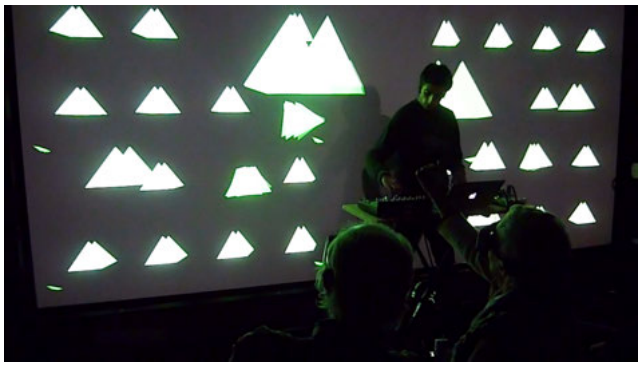


Figure 7: A shot from the performance: in the bottom right corner it is possible to see a spectator stretching his arm for interacting with the projected 3D particle system. When watched through shutter glasses the flat stereo projection is perceived as moving towards the audience.

5. EVALUATION

5.1 Questionnaires

Virtual_Real took place three times, allowing a total of 27 spectators to attend the show. We exploited the venues to give the audience a questionnaire, in order to collect data about the different aspects of the performance, as they experienced it. The questionnaire explored 5 specific evaluation areas; the first area, "General Evaluation", investigated the perceived similarity with other audio/visual performances previously attended by the audience. The second area was called "Perception", and dealt with the extent to which spectators perceived depth in 3D projections as opposed to 2D contents, while "Presence" area addressed interaction and immersion, including the sense of participation. In the "Transparency" area the global comprehensibility of the performance and the relation between artist's gestures and audio/visual output were investigated. The last area, "Specific Evaluation", dealt with the communicative role of 3D visuals, also compared to 2D sequences.

A total of 24 sentences (called also items) have been extracted from these 5 areas, and inserted into the questionnaire in a shuffled order; each sentence stated an observation regarding the related evaluation area. After the show spectators were asked to answer to what extent they agreed or disagreed with the sentences, choosing a number between 1 (completely disagree) and 7 (completely agree). Acquired data were then analyzed, focusing on central tendency, through median, mode and mean extraction, and on dispersion, calculating range across quartiles and standard deviation. In order to avoid predictable biases linked to the common astonishment generally felt during the first VR experience, before each show the audience attended a short training: a virtual environment was presented, in which each spectator had to complete some interactive tasks, touching and moving objects with head tracking and motion capture support.

Item score analysis showed amazingly positive results, which included most of the five evaluation areas (Figure 8); in particular remarkable results came from "Specific Evaluation", where almost the totality of the items scored median and mode values equal to 7: for example we can report that the 59% of the audience completely agreed saying that the sequences containing 3D visuals enhanced their involvement in the performance (S13). This item produced a median equal to 7.0, with a lower quartile equal to 6. Similar data (median equal to 7.0, lower quartile equal to 5) were extracted by the item stating "I preferred objects to come out from the screen" (S20), which was scored 7 by the 67% of the audience. Other items regarding the expressive power of visual interaction ("which helped to understand the artist's message", S9), the sense of participation to the performance (S7), and the perception of a world that grew "far beyond the physical boundaries of the room" (S17), scored very high median and mode values (6 or higher), less stunning results because of slightly stronger dispersion (e.g. mode percentage less than 50%), but extremely positive overall.

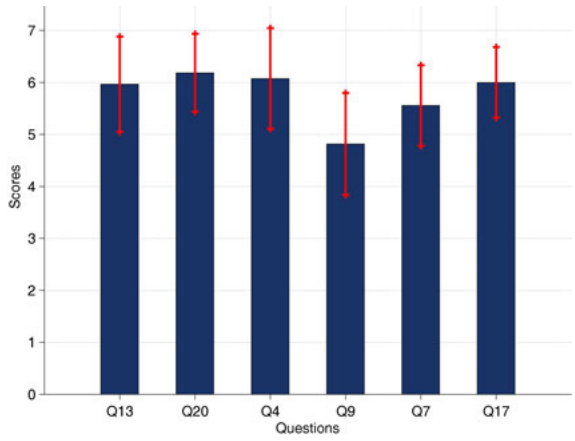


Figure 8: The graph shows in blue the mean value and in red the standard deviation value of each sentence level of agreement score. Sentences number 13, 20 and 9 come from "Specific Evaluation" area, 4 and 17 from "Perception" area, and sentence number 7 from "Presence" area.

In the last page of the questionnaire some blank space has been left to encourage spectators to leave comments, which turned out to be a very useful source of information. As expected, almost all comments were positive, remarking on spectators' astonishment and enjoyment already deduced from data analysis; some spectators also left interesting observations about features and aspects they liked the least during the performance, including important suggestions that could work to improve the proposed experience. Some spectators complained about the difficulty of understanding when audience interaction was available, as few parts of the choreography supported it, with limited effects over the environment; they agreed with the suggestion that more frequent and powerful interaction paradigms could spectacularly increase audience involvement. Others highlighted that whenever the 3D projections reached the borders of the screen or hit the body of the performer, sudden visual paradoxes temporarily interrupted the stereoscopic effect; they suggested using larger screens, taking extreme care that virtual objects never overlap with real stage elements.

5.2 Artist's Feedback

As the presented technology aims at supporting and inspiring artists and art production, we also invited the performer to write down comments and impressions, in order to understand to what extent he could exploit the platform as a strong communicative means.

Since the beginning of the collaboration USELESS_IDEA has been fascinated by the entanglement between the real and the virtual environment available on our platform. Furthermore the possibility to create an interaction, a dialogue, between these two worlds stimulated the artist to experiment new ways to communicate with the audience: mourning about the opacity and unclearness electronic music performances commonly suffer from, he underlined the expressive power of visual interaction, and the direct cause-and-effect relation caught by the audience. This brand new possibility strongly influenced the composition process too, opening new horizons for live composition and improvisation.

Two negative aspects were underlined: the lack of tactile feedback when handling virtual objects, and the disparity between performer's perspective and the projected viewpoint. According to artist's thoughts, these issues could invalidate the expressiveness of visual interaction choreographies, negatively influencing the overall result of the show; however, he added that, as for other common challenges in the domain of live performances, rehearsal sessions and a good support from technical team easily prevent these negative effects.

6. CONCLUSIONS AND FUTURE WORK

With this paper we presented a multimodal platform ideated to stage a novel kind of performances, called Hybrid Reality performances; thanks to VR technology, in the eyes of the audience artists are immersed within a 3D reactive environment, interacting with virtual objects to affect graphics and sounds. We named Hybrid Choreographies the set of rules that defines, for each performance, the meaningful relationship between artists' gestures and the surrounding audio/visual environment. These choreographies could include spectators' participation too, providing them with the possibility to transform each venue in a unique collaborative experience.

Theater, dance and music could be performed and even blended on this platform, to open the path to unpredictable artistic productions. The first example of Hybrid Reality performance was called *Virtual_Real*, and took place in our VR room as an interactive audio/visual concert; held by the electronic musician USELESS_IDEA, the show featured five music tracks, performed together with five Hybrid Choreographies, during which the artist created music both with real instruments and through virtual environment interaction.

After the show spectators were provided with a questionnaire, in order to collect data about the different aspects of the performance. Data analysis and comments from both the audience and the performer revealed a strong enthusiasm towards the platform capabilities, which really encourages us to continue artistic experimentation in Hybrid Reality environments. Thanks to these feedbacks we are now focusing on the use of dynamic shared viewpoints to provide also the artist with a meaningful visual feedback, and transparent screen technology to avoid virtual content occlusion.

To confirm these positive results, obtained in a well controlled technological environment, we are interested in moving on more conventional stages, like theatres and concert halls, through a portable setup complementary with local equipment and its infrastructure. Operating in this scenario would extend the possibility to attend the show to a much bigger number of spectators, in an environment exclusively ideated to host artistic performances. According to this necessity we are going to actively participate to NIME conference, performing live a Hybrid Reality music piece in which two performers create a progressive soundtrack along with the exploration of an interactive virtual environment.

7. REFERENCES

[1] 1024architecture. Euphorie. http://www.1024architecture.net/en/2010/02/euphorie-2/.

[2] J. Birringer. Empac.live.media+performance.lab. http://empaclivemediaperformancelab.blogspot.com/.

[3] A. Camurri, B. Mazzarino, M. Ricchetti, R. Timmers, and G. Volpe. Multimodal analysis of expressive gesture in music and dance performances. In *Gesture-Based Communication in Human-Computer Interaction*, 2003.

[4] ChunkyMove. Glow. http://chunkymove.com.au/Our-Works/Current-Productions/Glow.aspx, 2006.

[5] S. Fels and K. Mase. Iamascope: A musical application for image processing. In *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998.

[6] G. Kaiser and G. Ekblad. Audience participation in a dance-club context: Design of a system for collaborative creation of visuals. In *Proceedings of 2007 Design Inquiries*, 2007.

[7] A. Kaprow and J. J. Lebel. *Assemblage, environments & happenings*. H. N. Abrams, 1966.

[8] J. Lanier. The sound of one hand. *Whole Earth Review*, 1993.

[9] G. Levin and Z. Lieberman. Sounds from shapes: Audiovisual performance with hand silhouette contours in the manual input sessions. In *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression*, 2005.

[10] G. Levin, G. Shakar, S. Gibbons, and Y. Sohrawardy. *Takeover : Who's Doing the Art of Tomorrow?*, chapter Dialtones: A Telesymphony, pages 55–56. Springer, 2001.

[11] C. Martin, B. Forster, and H. Cormick. Crossartform performance using networked interfaces: Last man to die's vital lmtd. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, 2010.

[12] M. Matreyek. Glorious visions. http://www.ted.com/talks/miwa_matreyek_s_glorious_visions.html.

[13] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D, 1994.

[14] J. Samberg, A. Fox, and M. Stone. iclub, an interactive dance club. In *Proccedings of the fourth International Conference on Ubiquitous Computing*, 2002.

[15] R. Ulyate and D. Bianciardi. The interactive dance club: avoiding chaos in a multi participant environment. In *Proceedings of the 2001 conference on New interfaces for musical expression*, 2011.

[16] K. Vincs and J. McCormick. Touching space: Using motion capture and stereo projection to create a "virtual haptics" of dance. *Leonardo*, 43, 2010.

InkSplorer: Exploring Musical Ideas on Paper and Computer

Jérémie Garcia[1,2], Theophanis Tsandilas[1], Carlos Agon[2] and Wendy E. Mackay[1,3]

[1]INRIA, Univ. Paris-Sud
Building 490
Orsay Cedex
{garcia,fanis,mackay}@lri.fr

[2]IRCAM
CNRS UMR STMS
4 place Igor Stravinsky, Paris
carlos.agon@ircam.fr

[3]Stanford University
Computer Science,Gates
B-280
Stanford, CA

ABSTRACT

We conducted three studies with contemporary music composers at IRCAM. We found that even highly computer-literate composers use an iterative process that begins with expressing musical ideas on paper, followed by active parallel exploration on paper and in software, prior to final execution of their ideas as an original score. We conducted a participatory design study that focused on the creative exploration phase, to design tools that help composers better integrate their paper-based and electronic activities. We then developed *InkSplorer* as a technology probe that connects users' hand-written gestures on paper to *Max/MSP* and *OpenMusic*. Composers appropriated *InkSplorer* according to their preferred composition styles, emphasizing its ability to help them quickly explore musical ideas on paper as they interact with the computer. We conclude with recommendations for designing interactive paper tools that support the creative process, letting users explore musical ideas both on paper and electronically.

Keywords

Composer, Creativity, Design Exploration, InkSplorer, Interactive Paper, OpenMusic, Technology Probes.

1. INTRODUCTION

Composing music is a highly creative process, requiring both musical and technical skills. Within the past few decades, composers have been drawn to computers that offer powerful tools for specific tasks, such as *Finale* and *Sibelius* for editing scores, as well as full-scale programming environments, such as *OpenMusic* and *Max/MSP*. Composers use these tools to explore new musical ideas, generate novel sounds, and evaluate elements of a piece via real-time processing.

Composers are well-served with technology that helps them execute previously generated ideas. These tools can serve as a testbed, providing inspiration and the ability to test and assess different musical alternatives [2]. However, computer software is less effective for the earliest stages of the creative process, when the composer first struggles to represent a musical idea. Many composers still rely on pencil and paper for sketching partially formed ideas [12]. Coughlan [5] argues that, when expressing ideas, paper requires a lower cognitive load than software. A sketch can represent a complex, but as-yet incomplete idea: the details can be worked out later. Some sketches are rough and unfinished, others are carefully executed, such as curves that represent amplitude or other real-time processes [10]. Hand-drawn sketches are useful for working out a composition's structure, hand-written notes and annotations help the composer remember specific ideas. In fact, many composers design their own personal notations to represent and explore their musical ideas [12].

This paper describes our work with contemporary music composers to understand and provide technology that better supports the creative phases of the design process. We describe an initial study of how contemporary music composers at IRCAM use both paper and software tools. We present a framework for understanding their creative process, including activities on paper and in software. We then describe two design-exploration studies: a participatory-design study in which we worked with a composer on tools for paper expression and exploration, followed by the exploratory design of *InkSplorer*, an interactive paper application that links hand-written gestures to *OpenMusic* and *Max/MSP*. We used *InkSplorer* as a technology probe [11] to better understand the creative composition process and to explore how linking paper and software can better support innovation. We conclude with recommendations for the design of such tools and directions for future research.

2. RELATED WORK

We are interested in developing interactive systems that actively support the creative aspect of the composition process. Resnick et al. [16] propose a set of principles to guide the design of creativity-support tools. They emphasize the need for simple tools that encourage exploration of multiple alternatives and advocate using multiple tools, rather than just one. They argue that designers should begin with real-world observation and use participatory design for their development.

Composition software has a mixed record for supporting the creative process. In one in-depth study, Eaglestone and Ford [6] noted that an electroacoustic music composer had difficulty keeping track of electronic objects and navigating the various user interfaces. However, they also remarked on the experimental nature of his creative process and found that errors "often produce the most artistically interesting results". Amitani and Hori [3] explored how providing spatial music representations to the composer can improve creativity and Gelineck & Serafin [9] argued that computer tools that introduce some level of uncertainty may stimulate creativity.

A number of systems, including Xenakis' UPIC [13], Hyperscore[1], Qsketcher [1], Sonic Sketchpad [5], HighC[2] and Music Sketcher [20] were designed to take advantage of the power of sketching ideas, by linking drawings to music composition. These systems all use a mouse, graphics tablet or an electronic surface to draw musical forms on a computer screen. An alter-

NIME'11, 30 May–1 June 2011, Oslo, Norway.

[1] *http://www.hyperscore.com/*
[2] *http://highc.org*

native approach is *interactive paper* [14], which enables users to capture hand-written gestures on paper and transmit them to the computer. As with other tangible interfaces, interaction on paper is *space-multiplexed* [7] rather than *time-multiplexed*, as with a graphics tablet or a mouse. Users must view everything through a single window on a screen or tablet, rather than flipping through or spreading out different, potentially very large, sheets of paper. The physical representation of gestures on paper also affords exploration and offers both visual and computational reminders that can be quickly revisited, evaluated, and refined. The direct visual trace that the pen leaves on paper reinforces reflection on the task [17] and aids creativity.

Early interactive paper systems, e.g., *Digital Desk* [23], projected multi-media content onto paper or used a hand-held PDA to augment a biologist's notebook, e.g., *A-book* [15]. More recent systems [19, 22] use Anoto technology: a pen with a tiny video camera detects the precise location of each pen gesture with respect to barely visible dots printed on the paper.

Our previous research, *Musink* [21], used Anoto to help composers create and evolve personal notations on paper over time. We focused on initial expression of ideas, offering composers an extensible, gesture-based syntax with the freedom to incrementally create their own composition languages and link them to music software. Here, we focus on how users explicitly combine paper and software to explore ideas, with the goal of creating tools that support such exploration in both media.

3. OBSERVATIONAL STUDY

Before developing novel technology, we wanted to first understand the existing composition process, with particular emphasis on clarifying the early creative phases. We interviewed composers and watched them work as they expressed and explored musical ideas, either on paper or with software.

3.1 Method

Participants: We interviewed four advanced composition students from IRCAM, a center for contemporary music in Paris. All are experienced composers who have won prizes for their compositions. All have studied computer-assisted composition with software tools including *OpenMusic*, *Max/MSP*, and *Audiosculpt*. All are male, aged 30-40. We identify them by their initials: NM, AE, EM and MB.

Procedure: All participants were finalizing a composition intended for a soloist, with electronic elements. We asked them to bring this piece, plus their personal computers and any other related documents. Each interview was recorded and lasted approximately one hour. We transcribed and analyzed each interview, along with photographs or copies of their sketches and scores. We began by asking them to describe their current project and discuss how it evolved, in both paper and electronic forms. We asked Critical Incident-style questions [8] with recent concrete examples of how they addressed problems, followed by more general open-ended questions. At the end of each interview, we demonstrated a *Livescribe*[3] pen, which records sound with playback, as well as auditory and visual feedback. We asked them to brainstorm how such technology could assist their transition between paper and electronic representations or enhance their creative work in other ways.

3.2 Results

Expressing ideas on paper: Even though they are experienced users of composition software, all use paper to express their earliest musical ideas. Each has a unique way of working that varies in form and style. Some begin with blank paper and add musical scores or other graphical structures. Others develop personal notations to represent complex musical ideas or electronic processing (Figure 1). Their sketches include various graphical parameters, e.g. scale, color settings, orientation, envelopes, and thickness, which are mapped to musical parameters, often in an as-yet unspecified way.

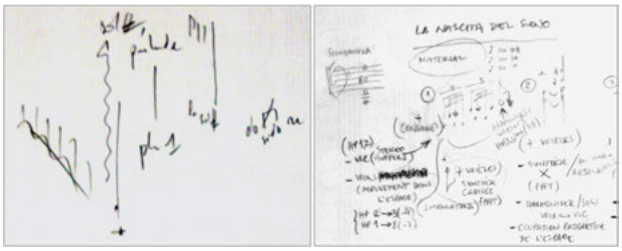


Figure 1. Left: Graphical representation of a piece (NM). Right: Hand-written scores and annotations (AE).

These composers distinguish sketches, which represent specific musical ideas, from underlying frameworks, which structure their ideas. For example, some composers redraw the musical staff; others use graph paper or specialized grids to lay out their ideas. This allows both flexibility and control when expressing concepts such as time, duration, pitch and density.

Exploring ideas on both paper and in software: After expressing their initial ideas on paper, composers move to an exploration phase, which involves both paper and the computer. NM described this as a tree: he generates and tests potential branches, successively accepting or rejecting them for the final composition. AE and EM use *OpenMusic* and *Audiosculpt*, combined with hand-written scores to experiment with ideas. NM and MB use *Finale* and *Sibelius* (music editors) to produce the final score, after first testing and printing some ideas. They also explore ideas using *OpenMusic*, exporting the results directly into a music editor or into *Max/MSP* as an event list to control electronic parts. Both NM and MB annotate printed or copied scores. AE and MB use real-time algorithms to control sound processing, AE, EM and NM use spatialization techniques and EM and MB use real-time synthesis.

Regardless of their technical expertise, all move back and forth between paper and software, sometimes drawing multiple curves on paper that they test in software, sometimes sketching an idea on paper that was inspired by a sound generated by the computer. Paper is clearly more flexible than software, demanding fewer constraints when expressing an ill-formed idea. For example, some composers use sketches to represent the structure of the whole piece or, like Marco Stroppa [12], use graph paper to draw extremely precise curves. When they move back to software, some paper-based representations get lost or must be translated into classical notation, which acts as a common language between paper and electronic representations. This runs counter to a Resnick's et al. [16] suggestion that "*creativity support tools should seamlessly interoperate with other tools*". Here, composers must shift between two methods of exploring ideas, forcing them to stay conscious of the medium and distracting them from the idea itself.

Representations evolve over time: The characteristics of drawings reflect different stages of the composition process. Figure 2 shows how MB's ideas evolve over time, as well as his use of paper and software. Figure 2a is a quick sketch, where the horizontal axis represents time, size correlates with amplitude and the orientation of the lines indicates transitions between notes. Figure 2b translates this sketch into a score, including a hand-drawn staff. MB does this to facilitate the transfer of the idea from paper to OpenMusic, which deals with curves and notes on a staff. Figure 2c is a printout of the corresponding musical object from *OpenMusic* which he has printed

[3] *http://www.livescribe.com*

on paper and added annotations, as explanations and reminders about what to try next. Figure 2d is the final score printed from *Finale*. He keeps both this score and his earlier hand-written sketches and printouts, as a record of his creative process.

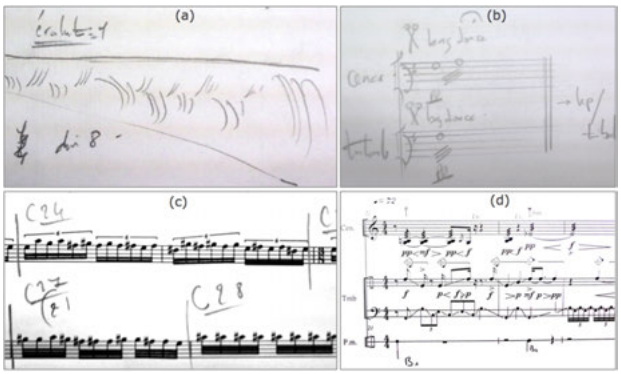


Figure 2. Evolution of musical ideas on paper (MB)

Conclusion: We found that composers engage in three main activities: *expressing* an initial idea, *exploring* it, and finally *executing* it in a composition. This cycle of expression, exploration and execution is highly iterative and occurs on both paper and in software, although paper-based activities occur earlier and end later. Figure 3 illustrates how composers use paper and software in parallel, without being able to truly integrate them.

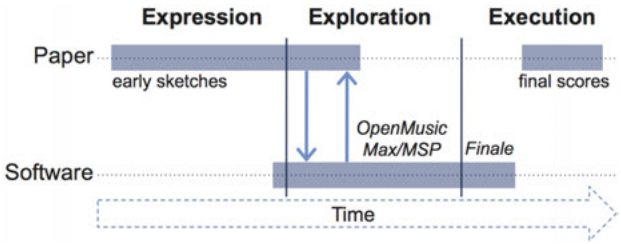


Figure 3. Composers work in parallel between paper and software, expressing, exploring and executing musical ideas

We were interested in what the composers thought about introducing a new technology, the *Livescribe* pen, into this process and asked them to reflect on how they might integrate it into their own work practices. They all wanted translations from hand-written notes into a musical editor, with the ability to modify or add details to scores printed from the software, ideally in a way that the software could then re-interpret. Although all were fascinated by the possibility of listening to parts of a score directly from the pen, they found the *Livescribe* pen itself too large and uncomfortable for daily use. All commented that they used pencils, not pens, and needed an eraser.

Based on these findings, we decided to conduct two studies to explore how interactive paper technology can aid the creative process. We wanted to offer composers the advantages of physical paper, with all its affordances, while also enabling them to benefit from the power of software tools. Our earlier *Musink* work focused on the initial 'idea expression' phase. Here, our goal is to support the middle exploration phase of Figure 3, more specifically, to help bridge the gap between paper-based and electronic composition activities.

4. PARTICIPATORY DESIGN STUDY

We used participatory design [18] to study how one composer explores musical ideas, with an emphasis on how interactive pens can enhance this process. Clearly, each composer has a unique composition process. We did not seek to find a generic solution, but rather to explore the design space and gain insights and ideas grounded in real-world composition activities.

4.1 Method

Participant: The composer (MB) had also participated in the first study.

Procedure: We first met with MB in a 2-hour participatory design session, followed by four shorter meetings over six months. We worked with a variety of different media including sketches on paper, a video prototype, and a Livescribe pen, used as a technology probe to capture data about his creative process and inspire ideas for new technology.

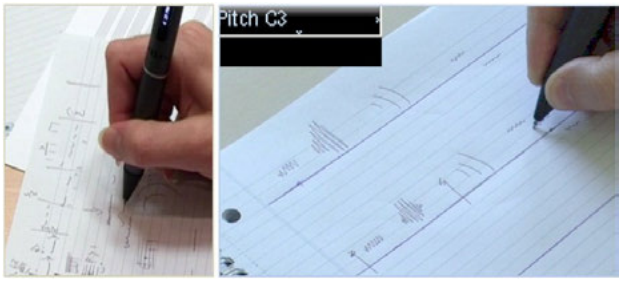


Figure 4. Left: MB explains his work process on paper. Right: Video prototype extract after a 2-hour design session

Livescribe pens run Java ME programs (*penlets*) and offer a range of functions, including auditory and visual feedback, audio recording and replay, interactive buttons and special areas printed on paper. For example, Figure 5 shows boxes with multiphonic tones for a saxophone piece that MB printed from *OpenMusic*. This let MB reflect on each sound while working on paper, using a penlet to replay sounds at will.

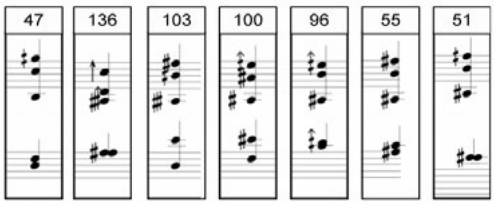


Figure 5. MB can tap on a box containing multiphonic tones and hear them from the pen

4.2 Results

MB offered a number of insights as to how he switches between expressing and exploring ideas, on paper and in software. He begins with sketches and gestures instead of classical musical notation so he can improvise and work *"at the speed of the thought"*. For him, providing live feedback from the pen would be too intrusive or distracting during the early stages of working out an idea. He is, however, interested in automatically translating paper-based gestures into classical notation that can then be interpreted by *Finale* or as an *OpenMusic* patch. This would save time and let him focus on expression rather than execution of ideas. He said the pen must capture as many data points as possible and he would find it 'unbearable' if the pen continuously notified him about what it had just recognized.

Design implications: Live interaction with the pen is not recommended for early expressive activities, but could provide the following useful functions during the exploration phase:

1. Record and play sounds by interacting with drawings or printed musical elements (as in Figure 5).
2. Evaluate and refine the result of drawings and gestures drawn on paper.
3. Define and modify rhythms and dynamics.
4. Restructure a piece by indexing different segments of the piece and exploring new structural alternatives.

We used a video prototype[4] (Figure 4) to explore how to implement some of these ideas. We created one space for the initial creation of ideas (gestures, musical symbols and drawings) and a separate "interaction space" that runs in parallel, along a common timeline. The latter was designed to be interactive and allow users to obtain information about their gestures, refine recognition and define rhythms. We explored additional interaction techniques to support this functionality including *Knotty Gestures* [22] to assign meaning, and physical transparent lenses [4] to refine the recognition of gestures.

5. TECHNOLOGY PROBE STUDY

We next investigated whether and how interactive paper could assist composers' exploration activities with *OpenMusic* and *Max/MSP*. Our goal was to enhance the computer-based exploration phase by providing additional physical space on paper for reflection, expression, evaluation and refinement of ideas. Based on our previous research [21], we also expected this technology to offer composers greater precision when defining musical parameters in a graphical form.

We developed *InkSplorer* to connect interactive paper technology to *OpenMusic* and *Max/MSP*. *InkSplorer* is a palette of tools, not a single prototype. This supports a technology probe [11] approach, in which our goal is not to validate a particular design solution, but rather to develop tools that composers can easily adapt to meet their individual needs. We hope to both gain new insights about the composition process as well as generate new ideas for designing interactive paper technology that supports the creative process.

5.1 InkSplorer

InkSplorer creates interactive paper with wireless Anoto ADP-301 pens that detect position and low-precision pressure. Pen data is sent to the computer via Bluetooth. Since drivers are not yet available for Mac OS X, we redirect pen data from a Windows 7 virtual machine to *OpenMusic* and *Max/MSP*. We use the OSC [24] communication protocol (fully supported by both *Max/MSP* and *OpenMusic)*. We created a library to manage storage and efficient retrieval of data so we can support real-time interaction with strokes on paper. The library uses the SpatialIndex library[5], an implementation of R-Tree, to store strokes, and was implemented as a Java external for Max/MSP. We also implemented patches and libraries in Common LISP for *OpenMusic* and Java for *Max/MSP*, to facilitate the integration of paper tools into composers' personal workspaces. The user interface in Figure 6 is a *Max/MSP* patch that lets users launch paper-aware applications and control pen configuration. The patch uses Jitter's OpenGL rendering to display incoming pen strokes.

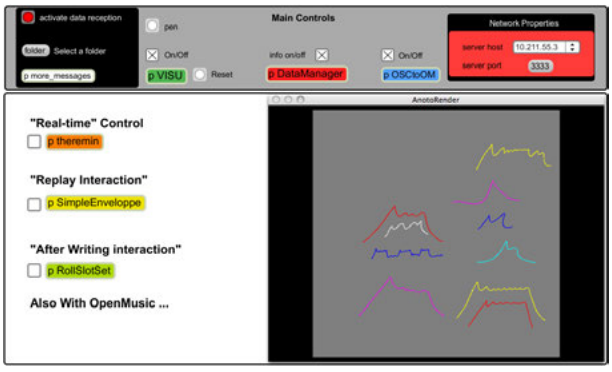


Figure 6. *Max/MSP* user interface for managing pen data

We developed a set of mini-applications of *InkSplorer* that integrate interactive paper into *Max/MSP* and *OpenMusic*:

1. A *Theremin*, controlled by moving the pen on paper.
2. A *Max/MSP* patch that maps pen strokes to sound envelopes (Figure 7, left).

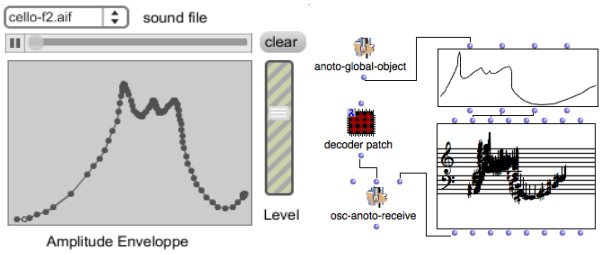


Figure 7. Mapping pen strokes to online graphical objects

3. *OpenMusic* patches that map strokes to BPF and BPC objects (Figure 7, right). Custom paper templates facilitate drawing and scaling of strokes.
4. *OpenMusic* patches that convert multi-strokes into musical objects using *maquette* [2] and custom paper templates.
5. Pen-drawing support for *bach*[6], a *Max/MSP* tool that enhances real-time processing with advanced musical notation. Duration and amplitude profiles of notes can be drawn on paper (Figure 8).

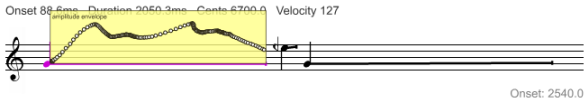


Figure 8. Defining a note's amplitude with pen data in *bach*

We created patches that detect and communicate various stroke properties: the x-and-y coordinates of each data point, data point density, pen pressure and time-stamps for each point. We can thus detect writing speed and variability throughout the duration of a stroke. From the user's perspective, *InkSplorer* provides a direct link between stokes on paper and the software.

Use Scenario: *MB is working on a piece for piano and real-time electronics. He has a clear idea for an electronic sound in his mind and captures it on paper in the form of a rough, abstract sketch with some text. He then creates an OpenMusic patch and proceeds to work out how to implement the sound. He inserts a BPF object to control the pitch range and turns to InkSplorer to explore different variations. He draws four curves on paper, singing the sound to himself as he draws. He taps on each curve and listens to the corresponding sounds produced by OpenMusic. MB likes the third best, but decides to change the final segment. He draws several slightly different curves on top of curve three and settles on the second variation. He adds an annotation to remember certain decision details, and circles the chosen curve, which stores it in OpenMusic. He also saves the original rough sketches and an OpenMusic printout in his notebook.*

5.2 Method

We conducted a series of mini-workshops with four composers at IRCAM, using *InkSplorer* as a technology probe to help them reflect on how to use interactive paper in their own work.

Participants: In addition to MB from the previous studies, three professional composers, KH, GL and MM, aged 31-52, agreed to test the *InkSplorer* prototype. KH, MB and MM had been interviewed in earlier studies [21] and were already famil-

[4] *http://vimeo.com/12853935*
[5] *http://trac.gispython.org/spatialindex/*
[6] *http://www.bachproject.net/*

iar with the basic Anoto technology. All were expert users of *Max/MSP* and *OpenMusic,* especially KH who had participated in the latter's development. GL and MM both teach computer-aided composition at IRCAM.

Procedure: We conducted a two-hour session with each composer, who brought his personal laptop and related documents, including musical scores, drafts of finished or in-progress pieces, and patches in *Max/MSP* and *OpenMusic*. All sessions were videotaped and later analyzed.

We first asked each composer about his background, professional activities, and experience and frequency of use of different music-composition tools. We then conducted a 30-40 minute semi-structured interview, focusing on how they represent and interpret curves and graphical forms, both in software and on paper. We asked for at least three specific examples and asked them to explain in detail how they worked out details, e.g. *"Describe the parameters this curve represents."* These interviews helped us to understand their work in context and identify concrete scenarios in which drawing curves on paper could be augmented with software functionality.

Next, we explained how to use *InkSplorer* and the mini-applications described above. Together with the composers, we selected examples from their work and imported their workspaces or parts of them to our laptop computer, where the pen drivers and *InkSplorer* had been installed. We successfully imported the *OpenMusic* workspace for three composers but not for KH, due to software version incompatibilities.

We asked composers to reflect upon how *InkSplorer* might change how they define, explore or refine musical parameters on paper and in *OpenMusic* or *Max/MSP*. We encouraged them to draw with the pen and use a 'think-aloud' protocol to describe its strengths and weaknesses. At the end of each session, we asked them to give us their reactions to *InkSplorer* as well as any suggestions they had for future designs.

5.3 Results

All four composers use *OpenMusic,* but only MB and MM use *Max/MSP* for composition. The other two use *Max/MS* for synthesis and interactive performance. These composers demonstrated diverse uses of curves to control various processes. For example, KH uses short curves to control an individual localized component of an algorithm or a synthesis process. Figure 9 (left) shows his use of a short curve to define a synthesis envelope or a pitch variation for granular synthesis. KH made a strong distinction between sound synthesis and music composition: For him, drawing curves to control synthesis, whether on paper or in software, is interesting, but he insisted that he is not a *"painter"* and does not use curves to compose music.

In contrast, MM uses long curves to control global properties of a piece or a section. Such curves are often more complex and more precise than short ones. Figure 9 (right) shows how MM uses long curves to control tempo variations in a 15-minute piece he composed for a short film.

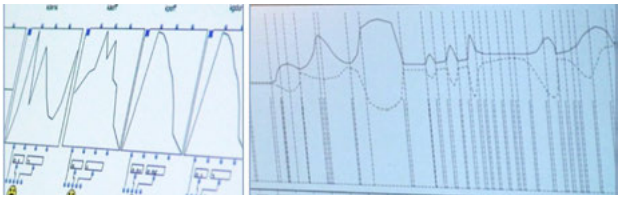


Figure 9. Hand-drawn curves control diverse processes

MM and KH use OpenMusic's maquette for spatial organizations of musical objects, controlled by temporal and graphical parameters. Reflecting on *InkSplorer*'s support for the maquette, the two composers showed examples from their work (Figure 10) that could be potentially produced by spatiotemporal mappings between paper gestures and maquette.

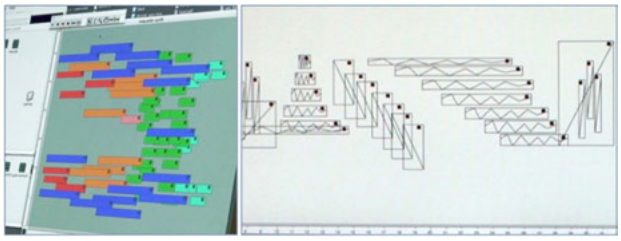


Figure 10. Spatial and temporal (x-axis) organization of musical objects (KH, MM)

The composers all chose to explore examples derived from their use of *OpenMusic*. The following issues concern both interactive paper in general and *OpenMusic* in particular.

Expressing ideas: Composers varied in how well paper helped them to express musical ideas. For GL, musical ideas reside in computerized patches and *InkSplorer* is only potentially useful for exploring these ideas faster. In contrast, MM feels that paper is simpler and more intuitive. For him, paper forms an *"analog"* space that provides more possibilities for expression than the computer, which he finds *"digital"* and constrained. MB finds the expressive power of both media to be similar, although he enjoys working with the pen more. He treats it as a musical instrument that involves physical movement of the body, a tangible sensation as the curve is drawn on paper: *"[I] use this pen just as I do an instrument. Here, I play the pen."*

Exploring ideas: MB and GL stated that *speed* is a major strength of interactive paper: it enables them to register multiple ideas and quickly assess their potential. MB feels that the pen saves time and helps him focus on the musical outcome rather than how to implement it. His hand-drawn gestures act as *memories* of sounds that can be returned to and replayed, even though the actual implementation resides on the computer. He also notes that computer screens have limited screen real estate whereas paper offers almost infinite space for exploring and "*The work is not lost in the computer"*.

Composers discovered interesting strategies for exploring ideas with *InkSplorer*. For example, MM drew several long curves on top of each other to evaluate different alternatives in the afore-mentioned composition, each providing incremental corrections (Figure 11, left). He used layers of curves to guide each refinement, explaining, *"It's a kind of guide that lets you correct it next time"*. In Figure 11 (right), MB draws variations of a short curve to control a 2-second sound synthesis.

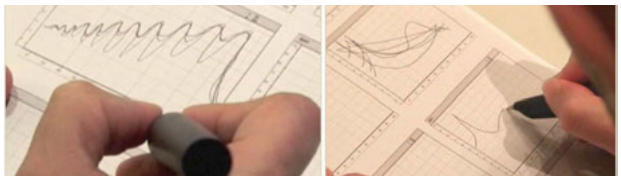


Figure 11. Reusing or refining curves (MM, MB)

Precision: Composers have different views about the relative amount of precision offered by paper and computers. MM feels that the computer is more precise because it lets him enter exact values, whereas data entry on paper is rougher. In contrast, GL finds that drawing on paper is more precise and lets him produce "more complex results". Finally, MB argues that although paper affords higher precision when drawing curves, it is not necessary for his compositions.

Design issues: The composers agree that integrating paper directly into existing tools, rather than creating a new interface, is the correct approach. However, they also want richer forms

of interaction. For example, MM finds it difficult to draw long curves without lifting the pen for a pause. He suggests that we let users easily connect segments together. Interestingly, only MM feels that capturing pen pressure or drawing characteristics such as pen angle are important, because these are essential in calligraphy. MB suspects that pressure might be useful, but would require practice to be controlled effectively. MM, GL and KH are particularly interested in using special pre-printed paper templates, particularly graph paper and musical sheets. In contrast, MB wants to create his own paper interface. The composers offered various suggestions for improving the pen design, including making them thinner, offering color, and supporting pencils or at least some form of erasure.

6. DISCUSSION AND CONCLUSIONS

Our goal is to design interactive systems that actively support the creative, exploratory phase of music composition. We conducted three studies to examine how professional composers combine paper and software tools. Study 1, based on interviews and observations of four composers, offers a framework for understanding the composition process, from early expression of ideas, to their systematic exploration and final execution. We found that composers explore both on paper and with software, as parallel, inter-related activities that they would like to better integrate.

Study 2 is a six-month participatory design study with one composer that explored how 'interactive paper' could better support his iterative testing of musical ideas. He argued that initial expression on paper is a 'delicate' phase and he does not want to be distracted by technology, i.e. live feedback that communicates state transitions and recognition errors. *Musink* [21], our first system, was designed explicitly to support the early creative phase, avoiding the interruption problem because data interpretation on first generation pens was delayed until it was uploaded to the computer. Although newer wireless pens offer real-time feedback, we recommend limiting this to later exploratory phases, when it is less disruptive.

Study 3 created *InkSplorer*, a pen-based composition tool that links paper-based and software-based to facilitate exploration of ideas. *InkSplorer* is actually a palette of mini-tools, which maximizes flexibility and supports both paper-to-computer and computer-to-paper testing and refinement of ideas. We tested *InkSplorer* with four professional composers. Their gestures on paper served as visual and computational elements that could be quickly revisited, replayed and evaluated, as well as layered and refined with new variations.

In future, we plan to more fully incorporate interactive paper into composition software such as *OpenMusic*. This will require tools that enable composers to define custom paper-based interfaces and richer, more powerful interactions with paper. We are particularly interested in gesture-based techniques such as our *Knotty Gestures* [22] and the use of portable electronic assistants [15] to aid the transition from symbols and gestures on paper to digital objects. Finally, we believe that *Musink* and *InkSplorer* are complementary and plan to integrate them in our future work.

7. ACKNOWLEDGMENTS

We thank the composers and researchers at IRCAM, especially Mathieu Bonilla, for their willingness to share their creative process with us. Thanks also to researchers at In|Situ|, in particular Stéphane Huot, for their help and insights.

8. REFERENCES

[1] Abrams, S., Bellofatto, R., Fuhrer, R., Oppenheim, D., Wright, J., Boulanger, R., Leonard, N., Mash, D., Rendish, M. and Smith, J. QSketcher: an environment for composing music for film. In Proc. *C&C'02* (2002), 157–164.

[2] Agon, C., Bresson, J. and Assayag, G. *The OM composers's book Vol.1 & Vol.2*. (2006). Collection Musique/Sciences. Ircam Delatour France.

[3] Amitani, S. and Hori, K. Supporting musical composition by externalizing the composer's mental space. In Proc. *C&C '02* (2002), 165–172.

[4] Bier, E.A., Stone, M.C., Pier, K., Buxton, W. and DeRose, T.D. Toolglass and magic lenses: the see-through interface. In Proc. *SIGGRAPH '93* (1993), 73–80.

[5] Coughlan, T. and Johnson, P. Interaction in creative tasks. In Proc. *CHI '06* (2006), 531–540.

[6] Eaglestone, B. and Ford, N. Computer support for creativity: help or hindrance. In *ARiADA*. (Vol.2, 2002).

[7] Fitzmaurice, G.W., Ishii, H. and Buxton, W.A.S. Bricks: laying the foundations for graspable user interfaces. In Proc. *CHI '95* (1995), 442–449.

[8] Flanagan, J.C. The critical incident technique. In *Psychological Bulletin*. (Vol.51, 4, 1954).

[9] Gelineck, S. and Serafin, S. From Idea to Realization - Understanding the Compositional Processes of Electronic Musicians. In Proc. *Audio Mostly* (2009).

[10] Healey, P.G.T. and Thiebaut, J.B.T. Sketching Musical Compositions. In Proc. *CogSci '07* (2007), 1079–1084.

[11] Hutchinson, H., Mackay, W., Westerlund, B., Bederson, B.B., Druin, A., Plaisant, C., Beaudouin-Lafon, M., Conversy, S., Evans, H., Hansen, H., Roussel, N. and Eiderbäck, B. Technology probes: inspiring design for and with families. In Proc. *CHI '03* (2003), 17–24.

[12] Letondal, C., Mackay, W.E. and Donin, N. Paperoles et musique. In Proc. *IHM '07* (2007), 167–174.

[13] Lohner, H. The UPIC system: A user's report. In *Computer Music Journal 10*. (1986).

[14] Mackay, W.E. and Fayard, A.-L. Designing interactive paper: lessons from three augmented reality projects. In Proc. *IWAR '98* (1999), 81–90.

[15] Mackay, W.E., Pothier, G., Letondal, C., Böegh, K. and Sörensen, H.E. The missing link: augmenting biology laboratory notebooks. In Proc. *UIST '02* (2002), 41–50.

[16] Resnick, M., Myers, B., Nakakoji, K., Schneiderman, B., Pausch, R., Selker, T. and Eisenberg, M. Design Principles for Tools to Support Creative Thinking. In *International Journal of Human-Computer Interaction*. (Vol.20, 2, 2006).

[17] Schön, D.A. *The reflective practitioner*. (1983). Basic books New York.

[18] Schuler, D. and Namioka, A. *Participatory design: Principles and practices*. (1993). CRC.

[19] Song, H., Guimbretiere, F., Grossman, T. and Fitzmaurice, G. MouseLight: bimanual interactions on digital paper using a pen and a spatially-aware mobile projector. In Proc. *CHI '10* (2010), 2451–2460.

[20] Thiebaut, J.-B., Healey, P.G.T. and Kinns, N.B. Drawing Electroacoustic Music. In Proc. *ICMC '08* (2008).

[21] Tsandilas, T., Letondal, C. and Mackay, W.E. Musink: composing music through augmented drawing. In Proc. *CHI '09* (2009), 819–828.

[22] Tsandilas, T. and Mackay, W.E. Knotty gestures: subtle traces to support interactive use of paper. In Proc. *AVI '10* (2010), 147–154.

[23] Wellner, P. Interacting with paper on the DigitalDesk. In *Commun. ACM*. (Vol.36, 7, 1993), 87–96.

[24] Wright, M. and Freed, A. Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. In Proc. *ICMC '97* (1997), 101-104.

Battle of the DJs: an HCI perspective of Traditional, Virtual, Hybrid and Multitouch DJing

Pedro Lopes Alfredo Ferreira J. A. Madeiras Pereira
Department of Information Systems and Computer Science
INESC-ID/IST/Technical University of Lisbon
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
pedro.lopes@ist.utl.pt, jap@inesc-id.pt, alfredo.ferreira@inesc-id.pt

ABSTRACT

The DJ culture uses a gesture lexicon strongly rooted in the traditional setup of turntables and a mixer. As novel tools are introduced in the DJ community, this lexicon is adapted to the features they provide. In particular, multitouch technologies can offer a new syntax while still supporting the old lexicon, which is desired by DJs.

We present a classification of DJ tools, from an interaction point of view, that divides the previous work into Traditional, Virtual and Hybrid setups. Moreover, we present a multitouch tabletop application, developed with a group of DJ consultants to ensure an adequate implementation of the traditional gesture lexicon.

To conclude, we conduct an expert evaluation, with ten DJ users in which we compare the three DJ setups with our prototype. The study revealed that our proposal suits expectations of Club/Radio-DJs, but fails against the mental model of Scratch-DJs, due to the lack of haptic feedback to represent the record's physical rotation. Furthermore, tests show that our multitouch DJ setup, reduces task duration when compared with Virtual setups.

Keywords

DJing, Multitouch Interaction, Expert User evaluation, HCI

1. INTRODUCTION

Through related work and previous research, we identified that standard DJ solutions have inadequate hardware requirements and are unable to cope with the rise of new features that modern DJs praise. Furthermore, they have high acquisition, maintenance and transportation costs; driving many professional DJs to look for alternatives, such as software DJing products. Although these applications include exciting features in terms of musical expression and extensiveness, they are bounded to a non-natural interaction scheme, derived from exercising indirect control via input devices instead of the gestural lexicon available in standard DJ solutions.

We classify DJ tools based upon their interaction and technological idiosyncrasies and identify three major setups: Traditional, Virtual and Hybrid. As multitouch technologies mature, they are applied in DJing, offering bimanual control of a virtual environment, providing DJs with digital sound processing advantages and natural interaction.

However, an evaluation of DJing interaction paradigms has never been performed. We present such an evaluation, aiming at understanding the virtues of multitouch in the DJ scenario. We focus on a novel Human-Computer Interaction (HCI) comparison of DJ setups, conducted with DJ experts, in which we compared our multitouch prototype against all three standard DJ setups. Ultimately we concluded on the adequacy of multitouch in the DJ context, allowing future researchers to build upon this set of hands-free interaction metaphors.

2. DJ SETUPS

In this section we present an overview of Traditional, Virtual and Hybrid DJ setups (Figure 1). The three DJ setups represent evolution stages of DJing tools, both in hardware and on interaction paradigm. We based upon the Traditional setup to understand the DJ mental model regarding physical hardware interaction. On the other hand, an analysis of the Virtual and Hybrid setups allows us to understand how DJs interact with digital tools. Furthermore, we review the interaction metaphors in recent controllers, as well as related academic research on multitouch controllers.

One must stress that there is a handful of published research on DJing across a multitude of platforms, thus the scope of this survey is limited to professional DJ systems and academic proposals that address DJing concepts. Therefore, DJ systems for casual mobile-phone operation or gaming/educational purposes are left out. Furthermore, since we aim for a comparison of the standard and multitouch setups, we left out other interaction paradigms that are not touch-based, such as Wearable [8], Haptic [2] and some Tangible [14].

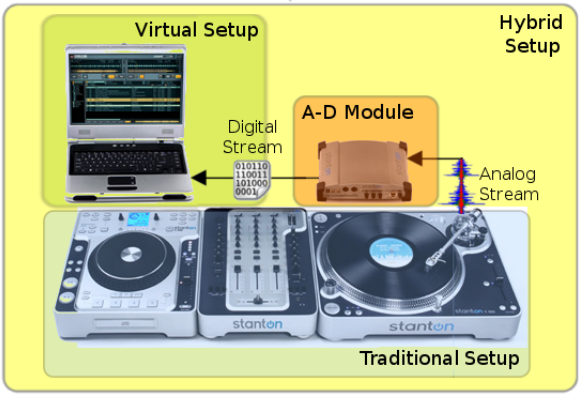


Figure 1: Relationship between the three setups.

2.1 Traditional DJ Setup

The Traditional setup is based on analogue devices, typically two turntables or CD-players and a signal mixer [7], as depicted in Figure 1. Throughout the last three decades, DJ's gestures and techniques have been strongly influenced by this setup.

In the Traditional users exercise direct bimanual interaction on the hardware, with instant visual and haptic feedback, which is an advantage [13, 11]; however, its dependence on complex and heavy equipment which often requires technical maintenance, is a drawback.

2.2 Virtual DJ Setup

Boosted by the rising popularity and increased performance of portable computers, the Traditional setup was virtualized into a software application, hence denoted Virtual setup. DJing applications in a Virtual setup provide digital audio processing, audio plug-in integration, unlimited tracks, and weightless storage environment. However, they are heavily criticized for their non-natural interaction, based on traditional input devices (mouse/keyboard) or dedicated hardware controllers, and also for their high learning curves, specially to users acquainted with traditional DJ gestures.

From an HCI perspective, most Virtual DJing applications offer a direct mapping of the Traditional setup, mainly because to make interfacing easier, the virtual controls resemble Traditional DJ gear setup (Figure 2). The user interacts with the turntable widgets to control the playout of songs, mixing them with virtual faders on a mixer-like widget. This visual trend follows the dual turntable metaphor, with a mixer in between.

(a) djay (b) Traktor DJ

Figure 2: Virtual DJ setup interfaces

However, Virtual setups face a serious HCI issue: they visually resemble the Traditional setup but do not feel as one. In this setup the user exercises indirect control [13, 11], not acting upon the interface, but operating at a distance, through a controller device.

2.3 Hybrid DJ Setup

To overcome the drawbacks of Virtual setups, the Hybrid setup was created, uniting Traditional and Virtual solutions. This gave DJs the possibility of using their traditional gestures over analogue gear to control a software application: by direct manipulation of the records, DJs are in fact controlling the digital audio playout. These systems depend on vinyl tracking to detect record position and acceleration. Although this solves the non-natural mapping problem found in Virtual setups, it also triggers the need for analogue equipment, known for its limited features and high acquisition, maintenance, and transportation costs. Furthermore, it has more limitations in terms of simultaneous playing tracks (usually two) when compared to the Virtual, since in the Hybrid all audio is controlled by turntables and needs to be routed back to the mixer.

Classifying a system as Hybrid may be misleading, because DJs may use full Virtual systems with accessory expression controllers without using computer input devices (typically mouse/keyboard). Our definition of Hybrid system embodies a setup that has at least one component found in the Traditional setup and one Virtual system; this categorization of the "Hybrid DJ setup" is also proposed by Bell [3].

2.4 Multitouch Controllers

Musical controllers have been around for quite a while, mainly due to the pervasiveness of the MIDI protocol - a *de facto* standard for audio control - but also because they provide a more natural interaction method for musical-related tasks [4]. Due to space restrictions, we focus on: the Lemur, a pioneer even amongst multitouch interfaces; the Reactable, with scratch[1] objects; the multitrack controller by Fukuchi, for uniqueness in gesture lexicon; and the Stanton SCS controllers, that include two finger touch-interactions.

2.4.1 Lemur

Lemur is a controller, although it has no physical controls (i.e., knobs, faders, IR beams) and everything is touchscreen based; all operations are performed on the touchscreen surface, as depicted in Figure 3. Lemur distinguishes itself from earlier controllers for its HCI contributions. The surface interface is fully customizable by the end-user, and new controls can be added, moved, resized and mapped into any OSC/MIDI message. This modular interface approach can be adapted by the user according to his needs, thus fitting any style of DJing. However, to accomplish customization, the user must operate through an offline application, running on a computer connected to the Lemur device. Although targeted at a larger usage spectrum, the

Figure 3: The Lemur device in touch control.[2]

Lemur can be used to perform some digital DJing tasks, namely because some GUI objects include faders and knobs, resembling the traditional gear. However, it is not designed for DJ gestures and there is no appropriate widget for the turntable-metaphor.

2.4.2 Reactable

In the Reactable [9] users can share control over the instrument by touching the surface or interacting with physical objects to build different audio networks. Each Reactable object represents a different audio concept, or component, with a dedicated function: generation, modification or sound control (see Figure 4(a)). Unlike the Lemur, which separates editing and playing modes, the Reactable combines both, creating a user-friendly, seamlessly integrated musical creation environment [10] - one more suited for synthesizer and audio processing, and not a traditional DJ tool. However, Hansen et al. have implemented a set of tangible objects to allow DJs to perform scratch gestures within this environment [6] and these are relevant to our research.

[1]Scratch is a DJ technique based on direct manipulation of record motion, that can be combined with fader movements.
[1]`www.jazzmutant.com`, accessed on 05/01/2011

(a) Collaborating [9] (b) Scratch Objects [6]

Figure 4: Reactable with Scratch Objects

In the Reactable framework, scratch uses three objects: the loop-player (controls audio), the movement-speed (controls turntable motion and speed) and finally the cross-fader (opens and mutes audio); with a combination of the three, a user can achieve the typical sound of scratching, as depicted in Figure 4(b). Hansen's evaluation [6] showed that as the test progressed, a Reactable expert became more optimistic and gained a deeper understanding; conversely DJs felt more pessimistic about the system, showing increasing discomfort with the control objects. Also, they identified that these systems cannot match the scratch experts' expectations of analogue turntable behaviour.

2.4.3 *Stanton SCS.3d and 3m*

Both SCS.3d and 3m are part of Stanton's "SC System Control Surface" product line. They are compact controllers with several two-finger touchable areas. Figure 5 shows that resemblance between the SCS.3d (a) and a turntable, or between the SCS.3m (b) and a mixer is not coincidental: Stanton expects to ease the user learning curve by mimicking the component design of a Traditional setup[3].

(a) SCS.3d (b) SCS.3m

Figure 5: Stanton SC System 3 [7]

Furthermore, it features some new interactions, not possible with typical DJ faders: touching directly on a slider's mark will make the value bump to that position - while in the "real world" a fader has to be manually dragged to the new position. Also, by holding one finger on the slider and tapping a new position with another finger will cause the cap to move to a new value for as long as that finger remains on the surface. When the second (upper) finger is removed, the slider will generate the value indicated by the first finger position.

2.4.4 *Multitrack scratch controller*

Fukuchi has proposed a multitouch-enabled device directed at scratching tasks [5], depicted in Figure 6, which enables the DJ to scratch several sources simultaneously, thus eliminating the time lost when users switch between various turntables. Fukuchi's Multitrack scratch controller is also highly effective in reducing space and component count. Fukuchi's metaphor does not follow the typical "revolving platter" found in turntables and in some DJ software, it uses a "moving waveform" metaphor. This metaphor is also used in Attigo[4] prototype, and can also be referred to as "conveyor-belt", because if the sound sample is looped, then the waveform reappears again in the beginning of the interface.

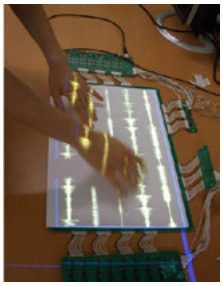


Figure 6: Multitouch scratch interface. [5]

In this multitouch interface the DJ interacts via direct manipulation of the waveform, making it move back and forth. Fukuchi also proposes a new metaphor, that enables DJs to perform record-crossfader combinations with just one finger, which increases scratching performance, but also generates a new "faderless" lexicon that was not easy to some DJs; thus raising the learning curve, as evaluations denote.

2.4.5 *Discussion on Multitouch Controllers*

Multitouch Controllers do offer new possibilities as Virtual DJing applications' controllers, so they stand out when compared to traditional mouse-based operations or button-based MIDI controllers. They allow improvements in two directions: increasing task performance by providing both bimanual interaction [11] and new interaction features (such as the aforementioned "slider jump"); and lowering learning curve by maintaining coherence with traditional DJ gestures.

With regard to visual feedback, these proposals have their own idiosyncrasies: Lemur and SCS are one-way slave controllers and therefore their visualization capabilities are often limited, and few DJing feedback can be shown to the user in realtime. On the other hand, Reactable and Multitrack scratcher do display the song's waveform, which is of value to DJs.

In terms of sensory feedback, touch-based controllers currently lack the tangible feel that Traditional and Hybrid setups can offer. Also, one must keep in mind that these controllers are application-driven, thus they do not offer interaction mechanisms for realtime DJ-specific tasks, such as adding new tracks, reorganizing setup (Lemur allows it in offline operation), altering connections between DJ components (audio re-routing), and so forth - all those tasks have to be carried out in the Virtual application, using a mouse/keyboard input device. Only the Reactable offers such possibilities, but then it is not driven by traditional DJ gestures.

3. MULTITOUCH DJ PROTOTYPE

To overcome the problems identified above, (non-natural interaction of the Virtual and the limitations associated with the Hybrid) we propose a multitouch interactive DJing application. The prototype was developed accounting feedback from four DJ experts (more details in [12]). The pro-

[3] `www.stantondj.com`, last accessed on 02/02/2011

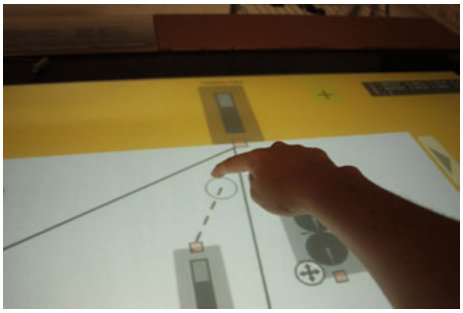
[4] `www.scotthobbs.co.uk`, last accessed in 02/01/2011

posed system merges the benefits of Virtual DJing applications with natural interaction found in Traditional DJing setups, rather than relying on typical input devices. Additionally, digital audio manipulation enables us to improve the DJ's performance, and also to exercise DJ creativity, by creating custom setups that are not possible in traditional live situations.

3.1 Interaction

Our interface is based on the following concepts: sound players, records, audio manipulators (volume faders, equalizer knobs, crossfaders, and so forth) and the relationships between these objects. These concepts are directly mapped into visual representations (of the objects) which the DJ can manipulate within a live performance, as depicted in Figure 7(a). All objects can be customized (moved, scaled, rotated) and linked to each other (see Figure 7(b)), allowing DJs to create a custom sound mixer, accordingly to their needs.

(a) DJ Mixing



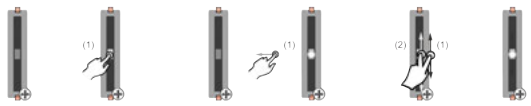
(b) Dynamic Audio Routing

Figure 7: Multitouch prototype in action.

The prototype supports the traditional gesture lexicon, and additionally our faders support new DJ-oriented features that altogether are not found in any previous work, namely: instant-jump, multiple-touch points and hold-down control. The instant-jump allows the fader to instantly jump without having to drag the fader cap manually.

In the multiple-touch points feature, the system not only registers multiple touches but also their order, returning to the previous position whenever the (last) finger is lifted. This allows instant kills on equalizers and fast crossfader switches.

The hold-down feature is the ability to control several faders with the same gesture. Figure 8 shows how to activate this feature: while touching a fader, the user can drag his finger around the canvas and still maintain control over the fader, up until that finger is raised, as depicted in Figure 8(b). If the user touches another fader with a new finger, that same behaviour is observed, thus the user can execute interesting motions, such as parallel control over



(a) First touch (b) Drag finger (c) Control faders

Figure 8: Controlling multiple faders with a single gesture.

two objects with the same hand, as depicted in Figure 8(c). Remarkably, our DJ testers took this feature even further; if one fader is rotated 180 degrees (up means volume 0), when the user moves both fingers in parallel up/down it is actually switching between those audio channels, without the need for the crossfader object.

Regarding record manipulation, the turntable widget mimics the platter's physical properties. Thus the user can expect the virtual record to behave as if it is under the force of the turntable motor, e.g., the record can be slowed down just by holding the finger in the label or, conversely, if one pushes the record forward, it will speed up until it reaches the normal torque. Both gestures are techniques that DJs use in analogue turntables to align songs together. Most techniques from traditional lexicon are also supported through the physical simulation that we provide, such as scratching, backspins and slip-cueing (hold and release the record instantly).

4. EXPERT USER EVALUATION

Tests were structured in three stages: a pre-test questionnaire to determine the DJ's profile and experience regarding multitouch devices; several DJ-oriented tasks; and, finally, an interview to get detailed information about interaction experience. With the users' permission tests were videotaped, application audio was recorded, interviews were transcribed from audio recording.

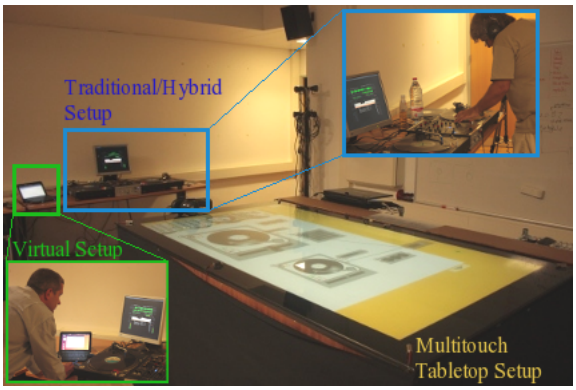


Figure 9: Multimedia room for Expert testing.

4.1 Apparatus

All tests were conducted in a closed environment, a large multimedia room with a 5.1 surround system, an 1.58x0.87m tabletop (LLP), a dedicated soundcard and DJ headphones. A video-camera recorded the test sessions, while a dedicated hardisk-recorder (with a pair of condenser microphones) captured audible output and user comments for later analysis. The room was fitted with all three standard DJ setups and our prototype. Figure 9 illustrates these setups, with

Virtual (Mixxx), Traditional (two turntables and a mixer) and Hybrid (Mixxx with vinyl tracking). Mixxx [1] was selected because none of the DJs had worked with it, thus levelling the test conditions regarding setup comparison.

4.2 Participants

Evaluation was carried out by ten DJs, four of them amateurs with two years of experience and six semi/professional DJs, with up to twenty years of knowledge. These DJs were not part of the aforementioned expert DJ panel to ensure that no previous knowledge would interfere with the test outcome. From our survey on DJ performance, we understood that different styles of DJing have specific application-requirements, and result in different DJ performances; therefore we included three Scratch, four Club, and three Radio-DJs in the testing group. Furthermore, all DJs were familiar with Traditional, Virtual and Hybrid tools, except the Scratch DJs that had never used Virtual setups.

4.3 Test Description

The tasks focused on mixing and beatmatching pairs of songs selected for their overall similarity, although with different tempi (ranging from 100-120 BPM). These songs were previously tested by consultant DJs to ensure that they had the same technical difficulty level and were indeed matchable. Songs were randomly selected from our song pool to guarantee that DJs could not to speed up the alignment task, in any test, by memorizing the correct pitch values.

Test-DJs were informed that no aesthetics judgement on the mix would be performed, as well as any skill-evaluation or score. In fact, DJs had to verbally inform us when they felt that both songs were aligned and the mix was completed. To further homogenize results we double-checked the video recordings of the tests; two different DJs (not part of the test-group) helped us in confirming the tasks' start and end points.

Each test session had five tasks, and a tutorial was given for Mixxx and for our prototype. The first four tasks aimed to mix/beatmatch a pair of songs in each setup, while the final task offered DJs a open session in our prototype, with songs of their choice. The first four tasks allowed us to develop a novel comparison between setups, and are denoted as: **V** (mixing on the Virtual setup); **T** (mixing on the Traditional setup); **H** (mixing on the Hybrid setup); and **Mt** (mixing on multitouch prototype).

4.4 Results and Discussion

From the test results we compute both the average and the standard deviation (σ) of the elapsed time for each task in every setup, as depicted in Figure 10. Our prototype's (**Mt**) result is better than that of the Virtual setup (**V**) with over less 100 seconds of elapsed time, proving that our setup is indeed more natural than the Virtual. But as expected, **Mt** took about 30 seconds more when compared with the Traditional (**T**) and the Hybrid (**H**), since the majority of our expert DJs has been using them for many years.

A detailed σ-comparison between all setups in Figure 10 validates not only the previous statement but also our group's DJ taxonomy (four Club, three Radio and Scratch), since DJs that classified themselves in a category got similar results (elapsed time) as others in the same class. **T** and **H** show a standard deviation of $\approx$26, while **V** ranks much higher, 76.89, because some users are not highly familiarised with this setup. This also shows that a separation of the results, Club/Radio vs. Scratch, is needed in order to evaluate the solution more precisely.

The average values shown in Figure 11(a) enable us to conclude that Club and Radio DJs operate quite well with

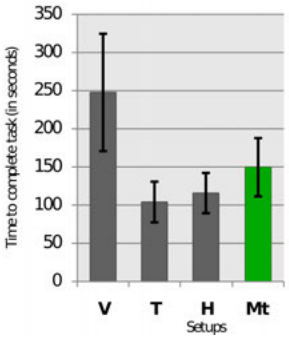
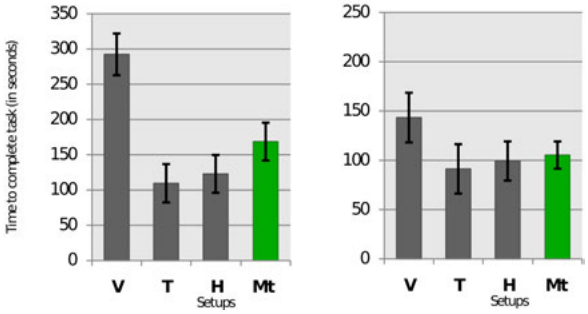


Figure 10: Average time needed time to complete the tasks for each setup.



(a) Club/Radio (b) Scratch

Figure 11: Separate analysis of Club/Radio (left) and Scratch DJs (right).

our solution, showing serious improvements when compared to the results in **V**. For Scratch experts, shown in Figure 11(b), we see that with **Mt** they only performed better than **V**, meaning that they are more efficient with direct record manipulation. Furthermore it seems that Scratch DJs perform faster in **V** than Club/Radio DJs because they tend to align and crossfade beats faster (as it suits their mixing style more accordingly). Indeed in **Mt** they exhibited a result very close to **T** and **H**, enabling us to conclude that the implemented physical simulation is worthwhile for those DJs.

In order to draw a final conclusion on the setups comparison, we must account the time differences for each user in each setup-pair, as depicted in Figure 12. It is easy to observe that the Virtual exhibited the worst results in any of

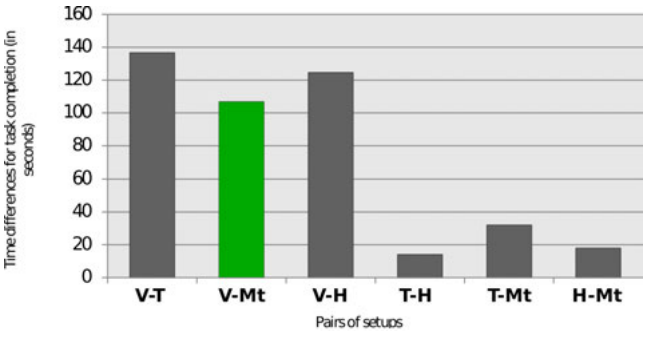


Figure 12: Time differences to complete tasks, between pairs of setups.

the comparisons, while in **V↔Mt** we observe our solution providing an average of 100 seconds of task improvement. This shows that touch support, bimanual and horizontal interaction help users to achieve better results with our setup than with Virtual Setups.

When comparing our setup to the Traditional/Hybrid, we collected optimistic results. **T↔Mt** and **H↔Mt** show that DJs mixed an average of 33.9 seconds faster in the Traditional and about 45.6 seconds faster in the Hybrid. This does not surprise us, since we are virtualzing the assets of the Traditional/Hybrid setup. The haptic feedback provided by touch surfaces is not good enough for Scratch-DJs, in particular when compared to the sensory feedback of the Traditional/Hybrid setups. Therefore, a multi-touch setup strikes a balance between available feedback and the digital benefits supported by the traditional lexicon.

Finally **T↔H** show a slight variation, because users tend to use the Hybrid solely through its traditional components, only using the computer for song selection.

4.5 DJ Comments

The overall feeling of DJs towards our multitouch proposal was promising. All users were keen to stress out the advantages of both bimanual interaction and multi-finger manipulation of the fader components, and also to denote how a tabletop environment offers a constant feedback during the DJ interaction, similar to the Traditional setup.

All users, including those who had no previous multitouch experience, mentioned that the interface was easy to use, and felt that the concepts were aligned the DJ's mental model. Manipulating objects around the canvas was recognised as a valuable feature for DJ users that want to exercise creativity in setup configuration.

5. CONCLUSION

To evaluate the adequacy of multitouch towards the DJing context, we tested DJ setups (Virtual, Traditional and Hybrid) against our proposal, with a panel of DJ experts; we also made a novel contribution to this subject area by cross-comparing all the standard setups.

The results suggest that our proposal can suit both expectations and needs of Club and Radio-DJs, but would fail against the mental model of Scratch-DJs due to the lack of haptic feedback of turntable motion. Tests show that Mt-DJing fared better than Virtual setups for all DJs, and task duration was reduced by an average of 100 seconds. As for tests against Traditional and Hybrid, multitouch solution slowed DJ tasks around 30 to 40 seconds. Our proposal has been quite favourably reviewed by DJ experts, which also contributed with additional comments, and have helped us in validating a set of gestural metaphors for the multitouch DJing context. From those we highlight: re-arrangeable interface, physical emulation of platter motion, dynamic routing between components and fader enhancements.

6. FUTURE WORK

Our work essentially studied touch-based interactions within the DJ context, leaving out many other interesting paradigms such as tangible or mixed reality scenarios. We plan to address these modalities, in order to analyse their contributions towards DJing.

Although the prototype was primarily targeted at medium/large multitouch tabletops, porting it to other platforms is possible; one can imagine how hand-held devices are exciting possibilities for DJ users that felt optimistic when mixing in our multitouch solution.

7. ACKNOWLEDGEMENTS

This work was partially funded by the European Commission's Seventh Framework Programme (FP7/2007-2013) through project MAXIMUS, grant agreement IST-2007-1-217039, and by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds.

8. REFERENCES

[1] T. H. Andersen. Mixxx: towards novel dj interfaces. In *NIME '03: Proceedings of the 2003 conference on New interfaces for musical expression*, pages 30–35, Singapore, 2003. National University of Singapore.

[2] T. Beamish, K. Maclean, and S. Fels. Manipulating music: multimodal interaction for djs. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 327–334, New York, NY, USA, 2004. ACM.

[3] P. Bell. *Interrogating the Live: A DJ Perspective.* PhD thesis, Newcastle University, 2009.

[4] N. Bryan-Kinns. Computers in support of musical expression. *ACM Computing Surveys*, 2004.

[5] K. Fukuchi. Multi-track scratch player on a multi-touch sensing device. In *ICEC 2007*, volume 4740, pages 211–218. Springer, 2007.

[6] K. Hansen and M. Alonso. More dj techniques on the reactable. In *NIME '08: Proceedings of the 2008 conference on New interfaces for musical expression*, 2008.

[7] K. F. Hansen. *The acoustics and performance of DJ scratching. Analysis and modeling.* PhD thesis, Kungl Tekniska Högskolan, feb 2010.

[8] K. Hayafuchi and K. Suzuki. Musicglove: A wearable musical controller for massive media library. In *NIME '08: Proceedings of the 2008 conference on New interfaces for musical expression*, 2008.

[9] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, NY, USA, 2007. ACM.

[10] M. Kaltenbrunner, G. Geiger, and S. Jordà. Dynamic patches for live musical performance. In *NIME '04: Proceedings of the 2004 conference on New interfaces for musical expression*, pages 19–22, Singapore, Singapore, 2004. National University of Singapore.

[11] K. Kin, M. Agrawala, and T. DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *GI '09: Proceedings of Graphics Interface 2009*, pages 119–124, Toronto, Ont., Canada, Canada, 2009.

[12] P. Lopes, A. Ferreira, and J. Pereira. Multitouch djing surface. In *ACE '10: Proceedings of the 2010 ACM SIGCHI international conference on Advances in Computer Entertainment Technology.* ACM, 2010.

[13] D. Schmidt, F. Block, and H. Gellersen. A comparison of direct and indirect multi-touch input for large surfaces. In *INTERACT '09: Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction*, pages 582–594, Berlin, Heidelberg, 2009. Springer-Verlag.

[14] N. Villar, H. Gellersen, M. Jervis, and A. Lang. The colordex dj system: a new interface for live music mixing. In *NIME '07: Proceedings of the 7th international conference on New interfaces for musical expression*, pages 264–269, NY, USA, 2007.

Designing Digital Musical Interactions in Experimental Contexts

Adnan Marquez-Borbon, Michael Gurevich, A. Cavan Fyans, Paul Stapleton
Sonic Arts Research Centre
Queen's University Belfast
BT7 1NN, UK
{amarquezborbon01, m.gurevich, afyans01, p.stapleton}@qub.ac.uk

ABSTRACT

As NIME's focus has expanded beyond the design reports which were pervasive in the early days to include studies and experiments involving music control devices, we report on a particular area of activity that has been overlooked: designs of music devices in experimental contexts. We demonstrate this is distinct from designing for artistic performances, with a unique set of novel challenges. A survey of methodological approaches to experiments in NIME reveals a tendency to rely on existing instruments or evaluations of new devices designed for broader creative application. We present two examples from our own studies that reveal the merits of designing purpose-built devices for experimental contexts.

Keywords

Experiment, Methodology, Instrument Design, DMIs

1. INTRODUCTION

Experimental methodologies within the NIME community have received greater attention in recent years. Both quantitative and qualitative methodologies for studying Digital Musical Interactions (DMIs) [17], deriving primarily from HCI, have been employed [35]. However, it has been noted [33] that the application of such methods has been limited, consisting of mostly informal user-studies of new and existing DMIs [10, 14, 21, 32, 33], or the creation of theoretical frameworks and taxonomies [4, 25, 27, 28]. Very few studies involving DMIs have employed purpose-built devices specifically designed as an integral part of the study. Rather, they tend to rely on either existing DMIs or devices designed for purely artistic purposes.

This is not to propose that this research has not been beneficial in testing or validating new DMIs or their underlying technologies, or that frameworks have not provided useful language and concepts for considering design. However, part of the reason for the limited reach of formal studies is that it is not obvious how to conduct them in musical contexts; transplanting existing methods from HCI will not always work. In addition, reliable generative frameworks are difficult to validate, especially in a creative domain that lacks easily specifiable evaluative criteria.

In this paper, we characterize the methodological approaches employed in studies of DMIs, as well as the particular choices of devices that have been involved in these studies. Subsequently we detail our experiences in incorporating the design of DMIs as an integral part of larger studies, which requires the designs to be functional and usable, but also to serve the particular goals of the overall studies. From these experiences we suggest new directions for the design of both novel research methods with DMIs.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. SURVEY OF METHODOLOGIES

We identify four main methodological styles of NIME studies. Although some certainly exhibit features of more than one, we believe it is an accurate characterization of the field, which reveals substantial space for novel methods.

2.1 Retrospective Taxonomies and Frameworks

Many studies aim to create comparative frameworks for the analysis of design approaches and consideration of features for novel design. These endeavour to provide universal criteria or dimensions upon which to consider all DMIs [3, 4, 25, 27, 28]; focus on specific features such as constraint [24]; or on particular contexts such as collaborative scenarios [5]. The meta-framework presented by Drummond [9] considers the variety of such prior approaches in terms of definition, classification and modelling of interactive music systems. Similarly, O'Modhrain [26] distinguishes between guidelines, frameworks, models and taxonomies, providing a review of prior work along these lines. Frameworks may be generative or analytical, but taxonomies tend to be retrospective. Indeed there are few examples in the NIME literature of generative applications of such studies; the literature has largely sought to categorize and situate existing or newly designed musical devices in the growing body of exemplars. A limitation of this approach, as noted in [4], is that the judgements of criteria that make up taxonomies are partly based on subjective assessments of the devices with little empirical evidence.

2.2 Evaluation of Newly-Designed DMIs

This approach involves post-build evaluation of a new device. These studies address particular issues or evaluative criteria by means of *a posteriori* examination in order to demonstrate utility, usability or functional improvement. This approach draws heavily on classical methodologies in HCI, where quantitative, empirical demonstrations of functional advantages are typically required. However, measurable operational gains are often difficult to demonstrate in creative applications, nor are they always relevant. As such, in the infrequent cases where evaluations of new DMIs are performed at all, they tend to be "informal" [33]. The evaluation of the "Pin&Play&Perform" [6], based on audience feedback, is a typical informal approach in NIME. While the evaluation of the Audiocubes [30] employed a more formalized observational study in an installation setting, it is unclear how this would translate to a performance context.

In contrast, Luciani et al. [23] explored formal qualitative and quantitative methods of how the "ergotic gesture-sound situation" contributes to the overall *playability, believability* and *presence* of virtual instruments.

Several authors have highlighted the shortcomings of applying quantitative usability measures from HCI to DMIs, (e.g., as proposed in [35]), arguing that their inherently reductive nature cannot adequately capture the richness and nuance of real performance scenarios [15, 33]. Stowell et al. [32] attempt to address these shortcomings by demonstrating a formal, qualitative approach to the evaluation of a novel timbre remapping system.

2.3 Evaluation of Existing DMIs

A posteriori evaluations have also been conducted on established instruments such as the Theremin, Radio Baton and Buchla Lightning, typically because of their widespread use and established performance practices. Evaluations tend to assess their ability to support specific properties or features such as expressivity, movement, timing accuracy and repeatability [7, 29, 36]; as well as to substantiate research methods for generalized application [14]. Existing consumer technologies appropriated for musical purposes, such as the WiiMote and mobile phones, have been evaluated in similar contexts [16, 20, 21].

2.4 Evaluations of Underlying Technologies

Another class of study focuses on evaluation of the underlying technologies and techniques of musical devices. An early study focused on establishing the most appropriate input transducers to realize specific musical functions [34]. Gelineck and Serafin [15] similarly investigated the differences between knobs and sliders as control interfaces in a physical modelling synthesis system.

3. AN ALTERNATIVE APPROACH

While this is by no means an exhaustive list of studies with DMIs, it highlights representative methodological approaches, many of which emphasize evaluation of inherent properties or features of existing DMIs. Our approach is distinct in that we are designing DMIs specifically to support research rather than evaluating features of an existing device. These questions demand more than a close examination of a device; rather, the device needs to fill a specific role within an experimental context. Our studies have thus necessitated designing purpose-built devices, which emerges as a very different task than creative design for a performance or artistic purpose [8].

We are not proposing a single and all-encompassing method for the study or design of DMIs. Rather, we present case studies that exemplify an evolving qualitative methodology that necessitates the design of novel devices in order to investigate the broader phenomena that underlie digital music interactions. Nor do we suggest that our designs do not afford artistic use. As we discuss below, we have found that for a variety of reasons the experimental contexts appeared to help performers develop and explore creative practices, although we stress that this is not our primary focus but a by-product of the methodology.

A small number of studies in NIME reflect aspects of our approach. The A20 [1] was created during a study on participatory design and evaluation, however the design emerged over the course of the study – it was an outcome rather than a prerequisite. Gelineck and Serafin's approach [15] is similarly allied in that the devices involved were designed expressly for their study. However, they consider these controllers to be "low-level interfaces" for the purpose of evaluating individual transducer elements, as opposed to performance-ready instruments. Perhaps most similar to our approach, the Spinotron was developed according to constraints and criteria dictated by the goals of a study investigating the design of continuous sonic feedback and quantitative methods for evaluating sound design [22]. In a perspective similar to that in [15], a purpose-built device is employed to reduce the influence of external variables that might affect experimental results. However, as in [15], the study authors did not consider the Spinotron as a musical instrument in itself: there was no established mapping between the Spinotron and the synthesis engine it drives. Indeed, one aim of the study was to evaluate the effects of controlled variations in this mapping [22].

4. CASE STUDIES

We present two case studies that illustrate our approach to designing DMIs in experimental contexts, highlighting how the context dictated the design process and the resulting artifacts. We show that this is a distinct activity from designing DMIs with artistic motivation, and that existing DMIs developed in other contexts would not have suited our needs. Furthermore, these experiences revealed important considerations for the design of DMIs in general.

4.1 One-Button Instrument Study

The purpose of this study was to explore the relationship between *style* and *constraint* [19]. We previously suggested constrained interactions – those in which the user's possible actions are limited physically, conventionally or perceptually – help spectators distinguish individual stylistic variations from the overall structure of an activity [18]. Ten study participants were to be given a week to practice in isolation with a constrained electronic instrument, after which they would individually play a short solo performance followed by a structured interview about their experiences.

4.1.1 Design Process

The main initial criteria for the instrument were to limit the number of controls and possible gesture-sound mappings. We considered studying existing constrained devices, but the purpose was to examine how style *emerged*; the burden of existing performance practices or conventions with known instruments necessitated the development of a novel device. We wanted the instrument to be minimal not just in terms of controls, but also in terms of suggestions of use.

Our design brief became to create a device that was purposefully minimal, that did not strongly invoke existing musical instruments or performance practices, but that still had an identity as a self-contained instrument. The device that emerged, the one-button instrument, is a "box that goes beep." It consists of a plastic project box with a single momentary button on its top surface. A tone of fixed pitch and amplitude is generated, and an LED is lit, for as long as the button is pressed. Other choices were guided by the desire to suggest a single action – pressing the button to play a beep – but not exactly how this should be accomplished. The rectangular shape of the enclosure lacked an indication of a particular orientation for holding it, or whether it should be held at all. The centrally-located button could just as easily be played with any part of either hand.

One unique aspect of the study context that drove the design process was a deliberate avoidance of considering an intended or normative "style" of use. This lies in stark contrast to typical design processes where specific usage scenarios are developed, and designers aim to direct users to these modes of operation. The design was further informed by the necessity to create 10 hand-assembled copies; the instrument had to operate consistently and reliably for each

participant but remain cost-effective. These factors exemplify why a device designed outside of the context of this study could never have been suitable, and reveal how the constraints of the study influenced not just the instrument's form but also the process of its design.

A first prototype had issues with pitch drift and timbre changes as the battery drained; a power switch was later added in order to preserve the battery over the week that the participants practiced with the instrument. This unplanned compromise resulted in an additional way for participants to manipulate the device, making it slightly less constrained than the ideal, but also gave rise to some of the most interesting and salient outcomes of the study.

We observed that participants found a variety of ways to play the primary feature of the instrument – playing tones with the button. Yet the perceived limitation also led to a surprising variety of techniques in which participants capitalized on "hidden affordances" [12] that were not conceived in the design process. These included: usage of the power switch to achieve timbre modification, exploiting mechanical noise in the spring-loaded button and manual filtering or modulation of the sound by cupping the opening over the loudspeaker. Although these "accidents" were ultimately beneficial in the context of our experiment, they reinforced the fact that even an apparently simple device may give rise to significant complexity.

4.2 Tilt-Synth Study

In contrast to the one-button instrument study, which focused on performer-instrument interaction, this wide-ranging qualitative study [11, 17] sought to understand spectators' experiences of performative digital musical interactions. According to the study's design, we presented spectators from a range of musical backgrounds videos of performances with contrasting instruments and conducted extensive interviews to compile and assess their experiences. Among the qualities we wanted to contrast were the degree to which the instruments would be familiar to participants and the extent to which the interactions would be understood.

4.2.1 Design Process

We chose the Theremin for one instrument in the study because we expected it would be known to some participants – the study deliberately included experts in the field of DMIs as participants – and even for the others, its gesture-sound mapping would be obvious. Conversely, the second instrument had to provide a baseline of no specific prior knowledge of the instrument. It had to offer sufficient complexity that some spectators would not understand how it worked, but that might be accessible to some.

These prerequisites constituted a challenging set of design imperatives. The device that emerged, the Tilt-Synth, is a standalone instrument built from segments of ABS pipe with a speaker located in one end. The synthesis comes from two PWM outputs of an embedded microcontroller activated by discrete switches on the opposite end. A second set of switches toggle between two modes of oscillation. In pitched mode, a two-axis accelerometer allows the performer to continuously control the pitches of the oscillators through the x-y tilt of the instrument, while two radial sliders control an amplitude modulation effect. In the second mode, tilt controls the bandwidth of a chaotic pitch stream.

The Tilt-Synth therefore combines large-scale physical gestures similar to the Theremin – facilitated by the continuous accelerometer control – with fine-grained discrete actions. The physical controls were designed to occupy all of the performer's manual actions and therefore minimize the ambiguity of address [2]. It was conceptually simple for a computer musician to perform and evokes obvious, large-scale actions with simple gesture-sound mapping even with minimal practice. However, the more subtle discrete controls and sonic non-linearity that results from the mode-switching provided some participants with ambiguous and incomplete understanding of its operation. The performer in our study, an experienced musician on acoustic and electronic instruments, rapidly developed a comfortable and distinct style of playing the Tilt-Synth that encompassed all of the features of the instrument.

5. DISCUSSION

Through the design and experimental use of the Tilt-Synth, we found that it takes very little complexity to "confuse" spectators. Even with a preliminary demonstration of the workings of the instrument, some participants in our study, including experts, had a poor understanding of how the device worked. Mode switching and chaotic oscillation were observed as significant contributors to perceived complexity; the fact that the performer could make the same physical action with two different results led to confusion of the gesture-sound relationship. Among some participants, this perceived complexity led to inflated evaluations of the performer's skill and experience with the instrument.

Other participants thought the performer's actions amounted to mere "button-pressing," suggesting the instrument was simple to master. Many concluded the performer must therefore possess intimate technical knowledge of the instrument, rather than bodily skill, in order to produce such a rich variety of sound. There was a similar perception that the performer was not fully in control of the sonic output, that he simply mediated some aspect of an automated system. Without an accurate understanding of the interaction, many spectators found it difficult to assess attributes of the performance such as skill or error. Yet some described a distinct aesthetic experience that drew more on the performative actions and dynamic sonic textures than on an actual understanding of what the performer was doing.

The Tilt-Synth study revealed a great deal about real, individual experiences we think are typical of NIME spectators. Only because the Tilt-Synth was purpose-built for this study were we able to examine the realistic situation in which a spectator watches a performance with a totally unfamiliar instrument and little contextual information. The lack of specific prior knowledge of the Tilt-Synth enabled us to investigate how individuals' diverse backgrounds, domain knowledge and aesthetic sensibilities interplayed with their perceptions of the performance. We could not have gained the same level of insight had we used an existing device or something other than a "real" performance-ready instrument. In contrast, in the same study the established performance practice, simple gesture-sound relation and pure tonal sound of the Theremin clearly drove spectators' expectations of what the performer was or should be doing.

With the one-button instrument, it became apparent that much of the stylistic variation was due to participants leveraging their expertise and drawing on their established musical practices. This revealed a great deal about style, as well as our design process; for some participants we succeeded in imparting an instrumental nature into the device, but one that was not so imbued with meaning as to suggest a singular or prescriptive mode of use or stifle their individual contributions. In this we observed an apparent contradiction to orthodox design principles from HCI. By deliberately not considering or prescribing usage scenarios, we enabled diverse users to develop meanings and styles that drew heavily on their own identities. But this was only possible because of the minimalistic nature of the design. Had

we tried to consider the needs, desires, expectations and experiences of our users it would have been impossible to realize such a minimal design, nor could it have supported the practices of experienced performers with idiosyncratic aesthetic sensibilities. This observation resonates strongly with the advocates for ambiguity and the support for multiple interpretations in design [13, 31].

Our experience with the one-button instrument further demonstrated that when conducting studies with real users in the real world, it is nearly impossible to account for every glitch in the instrument or for all the ways they can be appropriated. For such *in vivo* studies conducted over any length of time, it is therefore important to leave room for these kinds of anomalies or unintended artifacts in the study design. These are likely the very sort of "affective and creative" aspects of music-making that Stowell et al. [33] warn can be lost with reductive quantitative studies.

6. REFERENCES

[1] O. Bau, A. Tanaka, and W. E. Mackay. The A20: Musical metaphors for interface design. In *Proc. NIME*, pages 91–96, 2008.

[2] V. Bellotti, M. Back, W. K. Edwards, R. E. Grinter, A. Henderson, and C. Lopes. Making sense of sensing systems: five questions for designers and researchers. In *Proc. CHI*, pages 415–422, 2002.

[3] S. Benford. Performing musical interaction: Lessons from the study of extended theatrical performances. *Computer Music Journal*, 34(4):49–61, 2010.

[4] D. Birnbaum, R. Fiebrink, J. Malloch, and M. Wanderley. Towards a dimension space for musical devices. In *Proc. NIME*, pages 192–195, 2005.

[5] T. Blaine and S. Fels. Collaborative musical experiences for novices. *Journal of New Music Research*, 32(4):411–428, 2003.

[6] J. Bowers and N. Villar. Creating ad hoc instruments with Pin&Play&Perform. In *Proc. NIME*, pages 234–239, 2006.

[7] M. Collicutt, C. Casciato, and M. Wanderley. From real to virtual: A comparison of input devices for percussion tasks. In *Proc. NIME*, 2009.

[8] P. R. Cook. Remutualizing the musical instrument: Co-design of synthesis algorithms and controllers. *Journal of New Music Research*, 33(3):315–320, 2004.

[9] J. Drummond. Understanding interactive systems. *Organised Sound*, 14(2):124–133, 2009.

[10] S. Fels. Designing for intimacy: Creating new interfaces for musical expression. *Proc. of the IEEE*, 92(4):672–685, 2004.

[11] A. C. Fyans, M. Gurevich, and P. Stapleton. Examining the spectator experience. In *Proc. NIME*, pages 451–454, 2010.

[12] W. Gaver. Technology affordances. In *Proc. CHI*, pages 79–84, 1991.

[13] W. Gaver, J. Beaver, and S. Benford. Ambiguity as a resource for design. In *Proc. CHI*, pages 233–240, 2003.

[14] C. Geiger, H. Reckter, D. Paschke, F. Schutz, and C. Poepel. Towards participatory design and evaluation of Theremin-based musical interfaces. In *Proc. NIME*, 2008.

[15] S. Gelineck and S. Serafin. A quantitative evaluation of the differences between knobs and sliders. In *Proc. NIME*, pages 13–18, 2009.

[16] N. Gillian, S. O'Modhrain, and G. Essl. Scratch-off: A gesture based mobile music game with tactile feedback. In *Proc. CHI*, pages 234–240, 2009.

[17] M. Gurevich and A. C. Fyans. Digital musical interactions: Performer-System relationships and their perception by spectators. *Organised Sound*, 16(2), 2011.

[18] M. Gurevich, P. Stapleton, and P. Bennett. Design for style in new musical interactions. In *Proc. NIME*, pages 213–217, 2009.

[19] M. Gurevich, P. Stapleton, and A. Marquez-Borbon. Style and constraint in electronic musical instruments. In *Proc. NIME*, pages 106–111, 2010.

[20] C. Kiefer, N. Collins, and G. Fitzpatrick. Evaluating the WiiMote as a musical controller. In *Proc. ICMC*, 2008.

[21] C. Kiefer, N. Collins, and G. Fitzpatrick. HCI methodology for evaluating musical controllers: A case study. In *Proc. NIME*, 2008.

[22] G. Lemaitre, O. Houix, Y. Visell, K. Franinovic, N. Misdariis, and P. Susini. Toward the design and evaluation of continuous sound in tangible interfaces: The Spinotron. *Int. Journal of Human-Computer Studies*, 67(11):976–993, 2009.

[23] A. Luciani, J.-L. Florens, D. Couroussé, and J. Castet. Ergotic sounds: A new way to improve playability, believability and presence of virtual musical instruments. *Journal of New Music Research*, 38(3):309–323, 2009.

[24] T. Magnusson. Designing constraints: Composing and performing with digital musical systems. *Computer Music Journal*, 34(4):62–73, 2010.

[25] J. Malloch, D. Birnbaum, E. Sinyor, and M. Wanderley. Towards a new conceptual framework for digital musical instruments. In *Proc. DAFx*, pages 49–52, 2006.

[26] S. O'Modhrain. A framework for the evaluation of digital musical instruments. *Computer Music Journal*, 35(1):28–42, 2011.

[27] D. Overholt. The musical interface technology design space. *Organised Sound*, 14(2):217–226, 2009.

[28] G. Paine. Towards a taxonomy of realtime interfaces for electronic music performance. In *Proc. NIME*, pages 436–439, 2010.

[29] C. Poepel. On interface expressivity: a player-based study. In *Proc. NIME*, pages 228–231, 2005.

[30] B. Schiettecatte and J. Vanderdonckt. Audiocubes: a distributed cube tangible interface based on interaction range for sound design. In *Proc. Conf. on Tangible and Embedded Interaction*, pages 3–10, 2008.

[31] P. Sengers and W. Gaver. Staying open to interpretation: Engaging multiple meanings in design and evaluation. In *Proc. DIS*, 2006.

[32] D. Stowell, M. Plumbley, and N. Bryan-Kinns. Discourse analysis evaluation method for expressive musical interfaces. In *Proc. NIME*, 2008.

[33] D. Stowell, A. Robertson, N. Bryan-Kinns, and M. D. Plumbley. Evaluation of live human-computer music-making: quantitative and qualitative approaches. *Int. Journal of Human-Computer Studies*, 67(11):960–975, 2009.

[34] R. Vertegaal, T. Ungvary, and M. Kieslinger. Towards a musician's cockpit: Transducers, feedback and musical function. In *Proc. ICMC*, pages 308–311, 1996.

[35] M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from HCI. *Computer Music Journal*, 26(3):62–76, 2002.

[36] N. Ward, K. Penfield, S. O'Modhrain, and R. B. Knapp. A study of two thereminists: Towards movement informed instrument design. In *Proc. NIME*, 2008.

Crackle: A dynamic mobile multitouch topology for exploratory sound interaction

Jonathan Reus
STEIM
Achtergracht 19
1017 WL Amsterdam
jon@steim.nl

ABSTRACT

This paper describes the design of Crackle, a interactive sound and touch experience inspired by the CrackleBox. We begin by describing a ruleset for Crackle's interaction derived from the salient interactive qualities of the CrackleBox. An implementation strategy is then described for realizing the ruleset as an application for the iPhone. The paper goes on to consider the potential of using Crackle as an encapsulated interaction paradigm for exploring arbitrary sound spaces, and concludes with lessons learned on designing for multitouch surfaces as expressive input sensors.

Keywords

touchscreen, interface topology, mobile music, interaction paradigm, dynamic mapping, CrackleBox, iPhone

1. INTRODUCTION

Crackle is an interactive sound and touch experience for the iPhone that draws inspiration from the CrackleBox, the iconic "bent" touch synthesizer realized in 1975 by Michel Waisvisz at STEIM [7]. Crackle turns the multitouch surface of the iPhone into a changing sound landscape to be explored and shaped with the fingers.

The Cracklebox is likely the first commercially available portable, self-powered, audio synthesizer [2], and one of the first mobile music synthesizers to use the conductive qualities of the human body as a primary form of control. Modern multitouch smart-phones, heralded by many in the NIME community as a paradigm shift in digital music-making [4], are likewise self-powered, portable, and capable of generating a world of sounds through interaction via touch. It would seem that such a platform would be the ideal digital platform to realize an interactive experience inspired by the classic analog touch synthesizer.

One key challenge in developing Crackle was confronting the role of the human body in the sound-making process. The CrackleBox makes significant assertions about the body as an agent of control, both voluntarily (through applied touch) and involuntarily (by literally positioning the human body as part of the sound generating circuit). The iPhone, as a general purpose hardware platform with myriad levels of software abstraction, does not provide the low-level connection to its capacitive surface necessary to respond to the user's physiology. However, the touch-screen is a uniquely capable sensor for capturing applied touch. In developing his crackle instruments Waisvisz observed that physical effort exerted through human touch has an instantly recognizable way of shaping sound [7]. From this observation comes the core design philosophy of Crackle, to place the human body within an interaction paradigm that exposes the nuances of touch.

Figure 1: The CrackleBox.

2. THE CRACKLE EXPERIENCE

> "*It could be learned by playing by ear and developing experience and manual/mental skills instead of having to dive into a world of logic, functions, interaction schemes, electronic circuit theory and mathematical synthesis methods. One could play an electronic instrument in direct relation to the immediate musical pleasure of performed sound.*" - M. Waisvisz[7]

In Crackle we sought to re-imagine the expressive, explorative, and surprising qualities of the CrackleBox on the iPhone. In interacting with any functional object there are a set of rules which define the experience empirically. You push *that*, blow into *there*, and then *this* happens. Tanaka proposes that articulation of musical phrases is not typically executed by a single interaction, but rather a set of three interactions that work in conjunction to formulate a musical utterance: 1) Binary (on/off) 2) Basic parametric choices (choice of a string, or key) and 3) Expressive control (continuous control) [6]. This model roughly describes the interaction paradigm of the CrackleBox with the exception of 2). What follows is a description of the key empirical interactive qualities of the CrackleBox that we sought to recreate in Crackle.

2.1 Expressive Control Through Touch

In touching the fingertip-sized conductive leads of the CrackleBox, one is able to create large expressive sound gestures from minute, nuanced finger movements. A small twist of the index finger results in dramatic sweeps, another causes the device to chirp loudly. One could imagine a topological map of the sound world laid out over the surface of the instrument. In some places the curves of hills and valleys are dramatic, allowing wide sound gestures to be created by moving the fingers over a very small space. In other places, the landscape slopes gently, allowing the player to use a similar gesture to control the sound far more precisely. Crackle uses this topological approach to mapping touchscreen coordinates. The precision of the iPhone's touch screen allows for extremely subtle continuous changes in finger coordinates, facilitating the continuous control element described by Tanaka's model.

2.2 Exploration of New Spaces

To play the CrackleBox is to be in a constant state of discovery. As finger placement changes, so does the sound circuitry and thus the sound world of the instrument. It can even happen that you make a large gesture and get no sound at all. In Crackle, a digital system where such behavioral anomalies must be formalized, this exploratory paradigm is given as a conditional rule: when the user changes their arrangement of fingers beyond a certain threshold the topology of the interface must also change, presenting a new set of sound possibilities to the user (Fig. 3). Through play the user gains an intuitive sense of what movements are needed to produce what kinds of utterances. The changing landscape becomes a learned part of the playing process. Wessel proposes a formal control scheme for instrumental interaction in which he describes such exploratory learning using the metaphor of babbling [8]; the voice utterances which play a critical role in the development of speech in infants. Through its constantly changing interface topology, Crackle keeps the user in a state of babble. One is constantly relearning the interaction through exploration and intuition as the sound space changes and reveals itself.

2.3 Binary Articulation

The CrackleBox defies Tanaka's tripartite classification system in that there is no basic parametric control beyond the binary active and inactive. When nothing is touched, the box is silent. One articulates different notes by touching and releasing the playing surface. It would be reasonable, then, to consider an alternative classification system for certain instrument categories; one that collapses the binary and parametric choices into a single control type. Essl comes to a similar conclusion when analyzing the touch screen as a generic input modality. Observing that the multitouch screen, sans visual metaphors which segment the interface in non-tactile ways, offers two, not three, key interaction types: 1) two-dimensional local and moving coordinate sensing (continuous) and 2) explicit support for timed tapping on the screen (binary) [3]. With timed binary articulation such an important element of multitouch interaction, care was taken in Crackle to limit topological changes during such gestures for the sake of rhythmic reproducibility.

2.4 Open Interface Metaphor

The interface metaphor can be defined as a narrative framework in which to place the possibilities within the system into a context that is logical for the user [9]. Crackle, as a rule, favors a simple and suggestive interface metaphor over the musically denotative metaphors used in iPhone music applications such as Pocket Guitar (Bonnet Inc) and Pianist (MooCowMusic). This design choice echoes the progeny of the CrackleBox, which began with Waisvisz's desire to escape the connotations of religious music and western tonality inherent in keyboards used to control early analog synthesizers [7]. In Crackle, the iPhone's multitouch surface is visually segmented into six rectangular areas which give a hint where to begin, but do not enforce a particular interaction beyond presenting the touch surface as a playable object, encouraging the user to jump in, touch, and explore.

3. IMPLEMENTATION

The following is a discussion of how the interaction paradigm described in the previous section are implemented in Crackle. The interface is implemented conceptually through a combination of surface segmentation, mapping generation, dynamic remapping, and a pseudo-chaotic sound model. The combination of the individual segment mapping topologies with the unmapped interstitial space between segments creates a single complex surface topology for exploring the possibility space of the sound engine.

3.1 Touchscreen Interface Segmentation

At the bottom of it all is the iPhone's multi-touch sensor, whose Cartesian coordinate space is segmented into six rectangular sub-spaces. This initial segmentation is communicated on-screen as six rectangular touch zones; a simple visual queue that suggests where to begin touching. The user interacts with the interface by placing up to five fingers on the touch zones and by moving fingers within and between the zones. Aside from a small settings dialog button in the upper left corner of the screen, the entire screen surface is playable. More detailed information about the underlying topology could have been communicated through visualization techniques such as color-coded topological heat maps. However, it was decided not to implement such feedback in order to encourage this topological information to be discovered through touch and listening rather than visually.

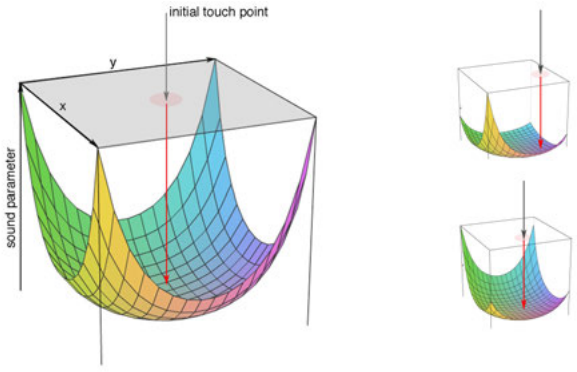


Figure 2: Graphs of conversion functions from segmentation coordinates to single control parameters. The graphs form a basin throughout much of the coordinate space and grow exponentially towards the edges.

3.2 Segment Mappings

Once segmented, the coordinate space of each segmentation is assigned a dynamically generated mapping to the sound engine's parameters. The mapping algorithms fall into four categories, based upon the target control parameters of the sound engine: period, modulation, period and modulation,

or dead-zone. When a mapping is assigned to a segmentation, unique conversion functions are generated based upon initial touch position and pseudo-random variation. The conversion functions reduce the two-dimensional parameter space of the segment to a single normalized sound control parameter. In the case where a segmentation simultaneously controls modulation and pitch, two independent conversion functions are generated. The dead-zone mapping is a mapping which controls nothing. The purpose of this mapping is to create an additional element of unpredictability and anticlimax during play. The analog Cracklebox also has this surprising quirk, but it bears repeating that with digital instruments surprises must often be formalized.

A 3D graph of the family of conversion functions illustrates the topological features of these mappings (Fig. 2). The graphs show a parabolic scoop whose shape flattens into a basin around the minimum point. The coordinate space of the flattened basin area is mapped to the less chaotic sonic qualities of the sound model, making these sounds easy to find and precisely controllable. As the finger approaches the far edges of the basin, the conversion function grows exponentially large, evoking sonically different and often chaotic results from the DSP algorithm. This mapping approach comes from personal observation of traditional instruments, where expected, "pleasant" tones are easy to find and control relative to more eccentric musical utterances. For example, a saxophone has a wide range of possible sounds, from the recognizable timbre of its stable tones to more abrasive honks and squeaks. The sax is built in a way, through the arrangement of finger keys and form factor, to deftly and easily command stable tones. While to master the more cacophonous, yet extremely expressive, part of the saxophone's sound world such as chirps, growls, portamento and quartertones, a performer must explore the eccentric edges of the instrument's interface. This was an apt approach for mapping Crackle's surface coordinates. The end result being that within segmentations Crackle gives the user a wide range of sonic expression but prioritizes the available sound world through ease of discovery and control.

3.3 Interstitial Space and Overall Topology

The interstitial space between touch zones is unmapped. However, a sample-and-hold strategy is employed on sound source parameters as a user moves her fingers smoothly from one touch zone to another. The slow de-zipping qualities of the DSP engine smooth any sudden jumps in parameter values as a new touch zone is entered from the unmapped space between the touch zones, making the user feel as if they are "jumping up to", or "falling in to", the scoop of the new area. Creating such discontinuities in the interface allows for complex sonic gestures which would not be possible using strictly continuous mapping.

3.4 Dynamic Mappings

In order to create the desired sensation of exploring an ever-changing sound space, the mappings of the six segmentations are dynamically remapped each time a finger is either added or removed from the touch surface. (Fig. 3)

During initial development, two algorithmic approaches were taken to generate the overall layout of six mappings: an evolutionary model, and a pseudo-random model. The evolutionary model generated a new mapping for each touch zone as needed, choosing the target control parameters and conversion function based upon the current mappings of the other zones. So long as a segmentation was still being touched, it's conversion functions were retained. When a segmentation was no longer touched its conversion functions would be removed from the configuration until it was touched again, at which point new functions were generated. Target parameters for newly touched segmentations were generated using an algorithm which first would analyze the existing configuration to ensure a functioning instrument that can produce sound (i.e. containing at least one pitch control mapping).

In contrast, the pseudo-random model generated a completely new mapping configuration for all six segmentations each time a significant touch configuration change occurred. The distribution of control parameters and conversion functions in the pseudo-random model was generated probabilistically, with additional checks to ensure a set of mappings that would create a functioning instrument.

The pseudo-random model proved to be the more successful of the two, based upon comparisons with the interaction experience of the CrackleBox. This is the algorithm used in the final implementation.

In addition, when a user moves fingers from one configuration to a new one and then back again (tapping the screen), the mappings of the first configuration are remembered and restored, providing the desired element of reproducibility.

3.5 Sound Model

The sound engine uses a system of squared sinusoids with a de-zipping algorithm applied to smooth changes in frequency and amplitude. This model was chosen because of its sonic similarity to the basic tone of the CrackleBox, and for the harmonics and pseudo-chaotic behavior found when given extreme parameter values as input, making this sound engine an especially apt choice for a wide range of sonic variety. The de-zipping algorithm is also deliberately slowed to create sounds similar to the characteristic pitch sweeps of the CrackleBox.

4. FUTURE WORK

In future work, we would like to apply the Crackle interaction paradigm to alternative sound models. We envision using Crackle as an encapsulated, general purpose interface for controlling any parameter space. Additional functionality could be implemented in the application to send normalized parameter values out via Open Sound Control to manipulate synthesis parameters of arbitrary sound models. A key question is discerning where the interaction experience ends and the sound model begins, and ultimately whether they can be divided at all. If a dividing line can be found, this implies that the application could be used as a general purpose controller for navigating sample material, granular clouds, or even lighting and visuals in a very characteristic way. Experiments and analysis would be necessary to determine if such a re-application would retain the same interaction experience and topological feel. Mapping has been covered extensively in the NIME literature, but the proposition of utilizing a mobile device as a black-box mapping and interaction system has yet to be fully explored. Such an encapsulated interface provides an attractive alternative to the limitations of simple one-to-one mappings without falling victim to the curse of programmability that hinders artistic mastery [1]. It is interesting to note that this approach flies in the face of the trends observed in many modern musical interfaces, such as the Monome and Syderphonics Manta, which promote an extremely open, programmable, architecture. At the very least, an encapsulated approach has potential as a method of developing widgets which could provide a canon of new multi-dimensional control metaphors for music on touchscreen-enabled computing devices.

5. DISCUSSION

Figure 3: Touchscreen segmentation and different finger configurations which would cause a remapping of the segments.

In designing Crackle, care was taken to preserve the role of the body's perceptual knowledge in interacting with the touchscreen. On-screen queues were provided to hint at the interaction metaphor, but not explicitly reveal it. Initial beta tests have suggested that soon after engaging with the application, visual feedback becomes a secondary modality to sound and touch. This attention switching, from visual to sonic and haptic, suggests that the visual modality at some point becomes less important to the interactive experience. Recent studies in cognitive science show that attention switching between sensory modalities such as vision and audition comes at a behavioral cost. Experimental subjects have exhibited slower perceptual judgements when constantly switching attention between modalities when compared to not switching [5]. This would imply that, as our beta testers began to actively inhibit one of the three sensory modalities (visual), they learned to minimize cross-modal attention switching and achieved a more immediate connection to sound production. Whether this suggests that the visual modality is in competition with touch and audition during musical performance is an interesting question warranting further experiments and empirical observations.

6. CONCLUDING REMARKS

Through developing Crackle we have learned a great deal about designing sonic interactivity for touchscreen computing devices. It is clear that creating an interesting underlying topology plays a key role in the success of using a touchscreen as a generic input modality for musical expression. The approach of mapping eccentricities of the sound model to less centralized (or extended) parts of the interface worked well, and we believe this is an approach that deserves further exploration in the design of new instruments for musical expression. It is also clear that limiting the visual metaphors of a touch screen to those which are not musically denotative encourages intuitive learning of an interface through babble, which can result in new and novel sound utterances. In treating the mobile touchscreen as a general purpose input modality, it appears that the key to a successful control scheme is a focus on topology, especially compound topologies. The mixing of continuous controls with discontinuities reveal new and novel ways of exploring a sound possibility space.

7. ACKNOWLEDGMENTS

The author would like to thank Frank Baldé for his assistance during the initial stages of development, Daniël Schorno and Dick Rijken for their numerous insights on instrument building and interaction design, and STEIM for supporting this work.

8. REFERENCES

[1] P. Cook. Re-Designing Principles for Computer Music Controllers: A Case Study of SqueezeVox Maggie. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME). Pittsburgh, Pennsylvania: Carnegie Mellon University*, pages 218–221, 2009.

[2] E. Dykstra-Erickson and J. Arnowitz. Michel Waisvisz: the man and the hands. *Interactions*, 12(5):63–67, 2005.

[3] G. Essl and M. Rohs. Interactivity for mobile music-making. *Organised Sound*, 14(02):197–207, 2009.

[4] N. Kirisits, F. Behrendt, L. Gaye, and A. Tanaka. *Creative Interactions - The Mobile Music Workshops 2004-2008*. University of Applied Arts Vienna, Vienna, Austria, 2008.

[5] S. Lukas, A. Philipp, and I. Koch. Switching attention between modalities: further evidence for visual dominance. *Psychological research*, 74(3):255–267, 2010.

[6] A. Tanaka. Mapping Out Instruments, Affordances, and Mobiles. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, 2010.

[7] M. Waisvisz. Crackle history. `http://www.crackle.org/CrackleBox.htm`, March 2004.

[8] D. Wessel. An enactive approach to computer music performance. *Actes des recontres musicales pluridisciplinaires, Lyon, Grame*, 2006.

[9] C. Zwick, B. Schmitz, and K. Kühl. *Designing for Small Screens*. AVA, 2005.

A principled approach to developing new languages for live coding

Samuel Aaron
University of Cambridge
Computer Laboratory
Cambridge
sam.aaron@acm.org

Alan F. Blackwell
University of Cambridge
Computer Laboratory
Cambridge
alan.blackwell@cl.cam.ac.uk

Richard Hoadley
Anglia Ruskin University
Digital Performance Labs
Cambridge
richard.hoadley@anglia.ac.uk

Tim Regan
Microsoft Research
Cambridge
timregan@microsoft.com

ABSTRACT

This paper introduces Improcess, a novel cross-disciplinary collaborative project focussed on the design and development of tools to structure the communication between performer and musical process. We describe a 3-tiered architecture centering around the notion of a Common Music Runtime, a shared platform on top of which inter-operating client interfaces may be combined to form new musical instruments. This approach allows hardware devices such as the monome to act as an extended hardware interface with the same power to initiate and control musical processes as a bespoke programming language. Finally, we reflect on the structure of the collaborative project itself, which offers an opportunity to discuss general research strategy for conducting highly sophisticated technical research within a performing arts environment such as the development of a personal regime of preparation for performance.

Keywords

Improvisation, live coding, controllers, monome, collaboration, concurrency, abstractions

1. INTRODUCTION

The Improcess project aims to create new tools for performing improvised electronic music in a live setting. The key goal of these tools is to structure the communication between the performer and a suite of concurrently executing musical processes. This fits within the broad genre known as live coding, where the performer writes and manipulates a computer program to generate sound. The Improcess project has started with a specific focus on a particular technical architecture and research strategy. The technical starting point has been to explore the potential for live coding performance combining domain specific music programming languages together with general purpose musical interface devices, such as the monome. Figure 1 shows a recent performance by the first author (shown on the left), in which a monome is used on stage together with a laptop running live coded software which is made visible to the audience via a large projection of the laptop's screen.

Figure 1: The $(\lambda - tones)$ performing live at the European Ruby on Rails conference in Amsterdam, 2010

The research strategy of Improcess has emphasised the creation of a cross-disciplinary team of collaborators rather than the more typical historical development of live coding systems, in which an individual developer works within an interdisciplinary context. We have also attempted to structure the project with a specific emphasis on the development of a reflective performance practice. The remainder of the paper discusses each of these aspects in turn: first our approach to experiments with the monome and with domain-specific languages (DSLs). Next, we address the architecture, the implementation, and the integration of these components into a personal regime of preparation for performance. We also reflect on the structure of the collaborative project itself, which offers an opportunity to discuss general research strategy for conducting highly sophisticated technical research within a performing arts environment.

2. THE END-USER APPROACH TO DOMAIN SPECIFIC LANGUAGES

In contrast to previous research in live coding, we start from a technical motivation related to the design and implementation of DSLs, languages specialised for creating particular kinds of application [15], and the design of languages for 'end-user programmers', people having no formal training in

NIME'11, 30 May–1 June 2011, Oslo, Norway.

programming [18]. The languages that have been developed for live coding in the past could all be classed as DSLs, although this terminology is not necessarily used. Some have also been designed according to principles that are well-known in end-user programming (EUP) - for example, the visual syntax of Max/MSP. However, many live-coders to date have been experienced programmers, rather than musicians who acquired coding skills as a way to advance their musical practice.

In the Improcess project, we believe that there is an advantage to be obtained by drawing on broader research addressing DSLs and EUP from contexts outside of music. We have noted in the past that live coding can be an interesting object of study for researchers in EUP [10]. However in this project we draw lessons in the other direction. In particular, we consider the 'cognitive ergonomics' of language design, and of programming environments. This leads us to consider the tools in the programming environment, alternative types of syntax (including options for 'visual' diagrammatic syntax as well as textual syntax), and also the underlying computational model. As an example of the diversity of tools, syntax and computational models that are considered in EUP research, the spreadsheet is often analysed as a DSL for the accounting domain. The tools are those for inspecting, navigating and modifying the visual grid. The syntax is the combination of the diagrammatic grid with formulae that refer to that grid, and the computation model is a form of declarative constraint propagation.

The Cognitive Dimensions of Notations (CDs) is a useful framework in which to consider the interaction of these different factors [12]. When designing new DSLs and programming environments, it is possible to optimise for different criteria - for example making languages that are very easy to change (low viscosity) or occupy small amounts of screen space (low diffuseness). All of these decisions involve tradeoffs, however, that can reduce the usability of the system in other respects. Duignan has in the past used CDs productively to analyse studio technology such as Digital Audio Workstations [14]. However, in addition to his concerns, we are also interested in abstraction gradient - is it possible for newcomers to start producing music without a large prior investment of attention? Some elegant computational models preferred by computer scientists are highly abstract, to an extent that discourages casual use. This is another trade-off that we consider explicitly.

Finally, our group has in the past explored the complex relationship between direct manipulation and programming abstractions [8]. In a performance context, direct manipulation allows an audience to perceive a direct relationship between action and effect. Programming languages, in our analysis, are essentially indirect - indeed, this is a fundamental tension of live coding. We hope that the monome can be used, not only as a device to trigger events (direct manipulation), but to modify the structure of musical processes in a way that maintains, enhances or interrogates this fundamental tension.

3. INTEGRATING CONCRETE INTERACTION

Many live-coders use peripheral devices, for data capture and control. However the software architectures tend to be influenced by conventional synthesis concerns, rather than being driven by the interaction devices, since live coding interaction is largely carried out via the laptop keyboard. We were interested in the opportunities that would arise by taking a live coding approach to a specific music interaction device, and using this as a fundamental element of the performance system being developed. We therefore made the decision to incorporate a concrete performance interface into the technical architecture that we are developing.

We are currently focussing on the monome, an interaction device consisting of 64 buttons arranged in an 8x8 grid on top of a box (a monome is visible between the two laptops in figure 1). Each button contains a concealed LED which allows them to be individually illuminated. Button presses are communicated via a serial link to the host computer, where they may trigger events within executing processes. The computer can illuminate the LEDs either individually, by row or column, or all 64 buttons. The monome only produces and responds to these simple data; there is no hard-wired or audio functionality. The monome's design is open-source, making it an ideal platform for free modification, rapid prototyping and experimentation [24]. Although the monome is described by its makers simply as a general purpose 'adaptable, minimalist interface', the list of published applications demonstrates that it is mainly popular as a music controller. The videos provided by the makers (for instance, [5]), most often adopt a control layout in which horizontal button rows control ordering in time and vertical columns either control pitch or trigger some selection from a variety of samples.

As a potential tool for the live coding performance context, we believe that the monome is complementary to the qwerty keyboard, offering a number of specific advantages. Its visible physical presence makes the actions and effects of interaction evident to the observer. Unlike conventional text programming languages it offers an extremely simple interface onto which musical patterns may be mapped as shapes. The set of button triggers enable responsive and direct communication with musical environment. Finally, the embedded LEDs provide feedback allowing the software environment to communicate aspects of its current state back to the performer and audience. This allows the monome to support a range interactive styles from a simple set of button triggers to a sophisticated low resolution GUI.

The majority of monome applications to date have been constructed using the visual dataflow language Max/MSP (56 out of the 68 applications listed on the monome community site). Max/MSP provides an excellent entry point to programming for musical end-users, but it lacks a number of important software engineering capabilities, which limit the ability of users to develop and maintain complex and sophisticated applications. For example, the visual format is not compatible with the version control tools that are essential for collaborative development. There is no framework to support the specification of unit tests needed for maintenance of robust systems. Finally, the only abstraction mechanism is a form of modularity through the creation of sub-patches. It is not possible to create abstractions capable of code-generation (i.e. 'higher-order-patches') which we believe to be a key capability for exploring new live coding practices. Our proposed architecture and implementation strategy provides an alternative framework that directly addresses these issues.

4. ARCHITECTURE

The Improcess architecture extends the traditional separation of audio synthesis technology from programming language implementation to explicitly consider a suite of bespoke interface components. This approach can support multiple alternative DSLs and external interfaces by providing a common run-time on top of which these clients may be designed and built. By sharing a common run-time environment these clients can therefore inter-operate allow-

ing a performer to create and combine a set of specifically designed interfaces for a given musical task. This separation of the traditional language implementation into system and client aspects allows for explicit design decisions regarding those system concerns that don't change from a given work or performance to the next (such as abstractions representing process and sound), from user concerns that must be changeable at any time, (such as the definition of new virtual 'instruments' and their particular control interfaces).

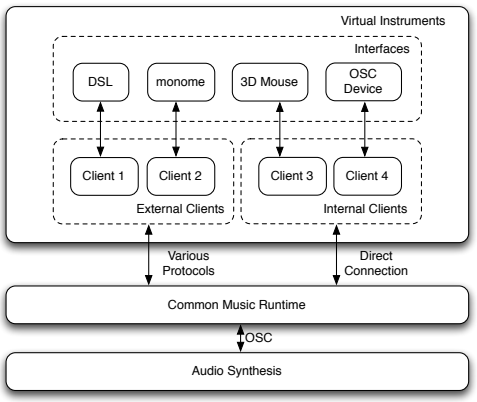


Figure 2: The Improcess architecture

The Improcess architecture consists of three tiers as illustrated in figure 2. At the bottom tier we have chosen to use the SuperCollider server as the audio synthesis engine. This provides an efficient, flexible and real-time-capable foundation, and allows us to focus on interaction and language design issues rather than audio processing. Controlling and manipulating the audio synthesis engine is the Common Music Runtime environment (CMR) which provides a suite of abstractions specifically designed for the creation and manipulation of musical processes and sound synthesis. Direct communication between the CMR and the SuperCollider server is defined in terms of Overtone [3], an open source project initiated by Jeff Rose with significant contributions by the first author.

Overtone, and the CMR, are implemented in Clojure, which was chosen because it provides a firm foundation for language experimentation and design. Clojure is an efficient functional language with an emphasis on immutability and concurrency as well as full wrapper-free access to the comprehensive Java ecosystem, highly flexible lisp syntax and meta-programming facilities that are currently best in class.

The top layer of the architecture is primarily meant for end-user interaction. This is via a suite of composable modular interfaces, which we call virtual instruments, at a level of complexity that can include complete DSLs. Each instrument has two aspects: an interface and a client implementing the required logic. Clients may either be implemented as a separate external process or as a concurrent extension to the CMR. The interaction with these instruments may either be via a conventional programming environment such as a GNU/Emacs buffer, or via a physical device such as the monome. These instruments then communicate with the CMR via the appropriate protocol in order to create and manipulate musical processes.

A frequent risk in the design of abstract architectures is the potential to disconnect from practical concerns within the application domain [9]. This can result in a design that may appear computationally powerful, yet not offer significant practical benefit for its intended purpos.e Another potential risk is the efficacy of the chosen technologies. For example, a given programming language might be able to express the desired operations yet not offer the performance semantics necessary to execute the operations within the required constraints.

We therefore took a strategic decision at the outset of the project to explore the interactive potential of our proposed software architecture by taking a number of established music applications for the monome, and re-implementing them within the proposed architecture using Clojure. Initially this has consisted of the re-implementation of monome prior-art including applications such as a sample looper, boiingg, Blinkin Park and Press Cafe. This also allowed us to explore some key technical parameters in the context of a mature interactive music application, rather than encountering them only while interacting with exploratory prototypes.

5. ABSTRACTION DESIGN

A useful and sufficient set of musical abstractions is a challenge for musical systems [7] and key to the success of the CMR. The programming abstractions found in SuperCollider are also widely available in other programming languages [19] and so it is clearly possible to reproduce equivalent semantics via a number of alternative methods. Hence we were left with the question of which musical abstractions to present to the user in the coding layer. Typical SuperCollider programs contain abstractions such as synthesisers, milliseconds, random numbers and routines. Would it be useful to also offer notions such as tunings, scales, melodies, counterpoint, rhythm and groove?

Such abstractions allow expressions that would have occurred in terms of the original complex sub-parts to be articulated more succinctly and accurately. Through redeveloping prior-art monome applications we were able to develop a vocabulary of abstractions pertaining to the monome shared across the group. We feel that this has provided a marked increase of the musical relevance of our discussions allowing much more subtle and precise notions of novel monome applications to be considered.

The main goal of the CMR is to present a suite of abstractions useful to the general task of building music performance processes. These can then be used and shared across a given set of clients to create new forms of instrument. The CMR currently supports many of the standard SuperCollider abstractions in addition to others found in alternative environments such as Impromptu's notion of 'recursion through time' [23] whereby recursive function calls are guarded by timed offsets.

An important aspect to consider with regard to abstractions is their symbolic representation. Within a procedural environment it is possible to automatically create new abstractions as part of the normal execution of the system. The ease with which this is possible is very much dependent on the syntactic structure of the representation. One of the main advantages of using a lisp as the syntactic framework is that it is very amenable to code generation. This is made possible because the syntax is represented by the basic data structures of the language (lists, maps and vectors in Clojure's case) and so generating and manipulating new code is as trivial as generating and manipulating these basic data structures.

As an example of this consider the CMR's synthesiser abstraction which comes directly from the notion of a synth in SuperCollider's server implementation. This is essentially a directed graph of unit generators such as oscillators and filters. Consider the 'bubbles' synth described SuperCollider's documentation. In listing 1 we have the SuperCol-

lider syntax for this synth which should be compared with Overtone's version in listing 2. Notice that conceptually they are very similar in addition to being represented with a similar number of characters. The Overtone syntax does not focus specifically on typing efficiency [22] but it is in a form that is relatively straightforward to generate automatically. This opens up a number of exciting possibilities where synthesiser definitions may be automatically generated by bespoke procedures driven by client interfaces. For example, we expect to create monome applications that allow the performer to both design and instantiate new synthesisers live as part of the performance.

Listing 1: SuperCollider bubbles representation

```
SynthDef("bubbles", {
  var f, zout;
  f = LFSaw.kr(0.4, 0, 24, LFSaw.kr([8,7.23],
                0, 3, 80)).midicps;
  zout = CombN.ar(SinOsc.ar(f, 0, 0.04), 0.2,
                  0.2, 4);
  Out.ar(0, zout);
});
```

Listing 2: Overtone bubbles representation

```
(definst bubbles []
  (let [root (+ 80  (* 3  (lf-saw:kr 8 0)))
        glis (+ root (* 24 (lf-saw:kr 0.4 0)))
        freq (midicps glis)
        src  (* 0.04 (sin-osc freq))]
    (comb-n src :decaytime 4)))
```

6. IMPLEMENTATION ISSUES

Music processing architectures often introduce subtle and demanding engineering issues relating to the management of processor load, timing mechanics, and latency. A detailed discussion of these engineering issues is not appropriate to this overview paper, but in this section, we record some of the general implementation challenges that we have encountered while developing the CMR, both as a warning to newcomers, and to confirm findings reported by other developers in the past.

6.1 Event stream architecture

One of the main roles of the CMR is to convert performer actions such as button presses into generated audio that is specified in terms of musical processes. From an implementation perspective a key consideration is the internal representation of these actions and processes, particularly with respect to time [16]. Existing music synthesis architectures are often event based, but as already noted, they do not support higher-order descriptions to the extent offered by functional programming languages i.e. function composition. But in a functional language, it might also be considered more natural to represent action in terms of a function. We therefore had to decide whether to use functions or events as the "first class" internal representation of musical actions.

A key concern within our process-based model was support for thread concurrency. Function calls are typically a synchronous mechanism, with function calls executing within the current thread. Events are typically asynchronous, resulting in concurrent execution by a separate thread. For this reason, we chose an event model of musical actions, to better support our overall concern with concurrent processes in music. This also plays to Clojure's strengths given its novel concurrency semantics which are currently state of the art relative to other functional languages. In particular, events can then be represented by immutable data structures. This means that they may not be modified once created, allowing them to be freely passed around from one thread to another without risk of being modified by one thread whilst being simultaneously used a second - a common cause of error in concurrent systems, and one that would be undesirable in a live coding context.

Events also provide an intuitive computational approach to combining musical actions. For example, we may combine a series of events into a stream which is ordered in time. We may then consider flowing the stream of events through a series of functions in a similar manner that we may wish to flow an audio stream from an electronic guitar through a series of effects pedals. As in other audio processing systems such as Max/MSP, this patching together of streams of events through functions also opens us up to the possibility of forking and merging given streams. We can therefore uses a source stream to create a number of new streams which contain an identical series of events but continue on their own path. One application of this would be to generate a chord with each note played by a different synthesiser yet triggered by one root event.

For example, consider Figure 3 which represents a simple event stream setup for sending basic monome keypress events to two separate synthesisers A and B. In this example we are taking the basic events from the monome, and adding them to a buffer which then forked to two separate streams - one which is connected directly to synthesiser B and another which is mapped through a function to modify the parameters of each event and then sent to synthesiser A. This approach is extremely flexible allowing the performer to define arbitrary modifications to a given input stream. Provided the performance semantics of the mapping functions is within acceptable boundaries the end to end latency of the event stream should be acceptable for live performance.

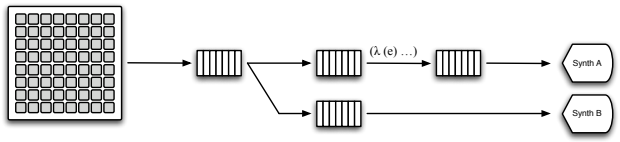


Figure 3: A typical monome event stream configuration

6.2 Performance and timing

Prior to the Improcess project the first author had implemented a monome application framework, 'monomer', using a generic programming language [4]. Monomer was intended as a general purpose toolkit for building monome applications which interfaced with external audio synthesis software via a basic MIDI interface. The result was sufficiently capable to build tools such as 8track (a Roland X0X-style MIDI step sequencer with 8 instruments / tracks [2]), but monomer suffered from two technical problems that posed considerable obstacles to building more sophisticated applications. First was the issue of performance. Even applications with minimal logic used an excessive amount of CPU. Whilst this might be acceptable for basic applications, it meant that adding any more complexity soon saturated the machine. Secondly, monomer's timing mechanics were built on top of Ruby's kernel method sleep. However variable delay in the process scheduler meant that the actual sleep time was greater than specified, and resulted in poor synchronisation with external rhythmic sources.

These experiences informed the Improcess architecture and implementation. Precise timing of event triggering is

handled by SuperCollider, which uses a prioritised real-time thread rather than Ruby's non-realtime sleep operation. Clojure's execution performance far exceeds that of Ruby, and is able to further optimise particular code paths by adding type hints. Within our new architecture the main performance issue currently faced is the variable latency of message streams within the CMR. This is not always acceptable within a performing context (it may result in stuttering or jitter of audio output, or delayed response to button presses). This is a general issue of modern computing architectures due to the fact that time is often abstracted away [17]. In our specific case it may be due to low level mechanisms such as the JVM garbage collector (GC) halting all executing threads during the compaction phase or intermediate IO buffers interrupting the flow of events. In most cases these issues are resolved by fine tuning the GC, scheduling events ahead of time where possible and the removal of blocking event calls. However, this is not always an option - particularly in a situation whereby the performer wishes to trigger an immediate musical event.

7. DEVELOPING A PRACTICE REGIME

A key concern of this project has been to maintain a research focus on live coding as a performance practice, rather than simply a technical practice. An explicit goal of the research was for the first author, who was already a highly competent software developer and computer science researcher, to acquire further expertise as a performer. We build on research previously reported at NIME, including Nick Collins' (alias Click Nilson) discussion of live coding practice [21] and Jennifer Butler's discussion of pedagogical etudes for interactive instruments [13]. Butler's advice could certainly be applied to monome performers, as a 'method' to develop virtuosity on that particular controller. However, the notion of virtuosity for a programmer (or composer, as in Nash's work [20]) is more complex.

As Click Nilson reports of experiments in reflective practice by his collaborator Fredrik Olson:

> *"'I [Olson] feel I'd have to rehearse a lot more to be able to do abrupt form changes or to have multiple elements to build up bigger structures over time with. I sort of got stuck in the A of the ABA form.' Yet we also both felt we got better at it, by introducing various shortcuts, by having certain synthesis and algorithmic composition tricks in the fingers ready for episodes, and just by sheer repetition on a daily basis."* [21], p. 114.

Nilson suggests a series of live coding practice exercises, which he warns must be approached in a reflective manner. As he notes,

> *"it would be a Cagean gesture to compose an intently serious series of etudes providing frameworks for improvisation founded on certain technical abilities"* (ibid, p. 116).

Despite Click Nilson's love of Cagean gestures, we agree that there is potential for a reflective approach to live coding practice to inform the development of etudes for the live coder. Others have noted this, for example Sorenson and Brown [22] describe the problem of being able to physically type fast enough, and suggest that technical enhancements such as increased abstraction and editor support are the most important preparations for effective performance.

Our collaborative project between senior computer scientists and music academics has encouraged a perspective that extends beyond virtuosity as a purely technical accomplishment, to the artistic engagement of a practised performer with a live audience. This resulting change with respect to the intellectual scope of the first author's prior experience of DSL research can be compared to recent attention in computer science, and especially within HCI, to the value of a "critical technical practice" [6] that integrates critical and philosophical perspectives into computing. In the case of our own work, we consider performance as a category of human experience that does not arise naturally from technical work, and hence must be an explicit focus of preparation and reflection. We might call this a 'Performative Technical Practice', by analogy to Agre's work.

As noted by Nilson, Butler and Sorenson, while all musicians prepare themselves through regular practice, the analogies between conventional musical instruments and programming languages are far from straightforward. Live coding is, in many ways, more similar to musical composition than to instrumental performance. Yet it seems Quixotic to distinguish between 'practicing composition' and 'doing composition'. A composer practices composition by doing more composition. If regarded in that light, any coding might be regarded as practice for live coding. However, we did not believe that this offers adequate insight into the special status of live coding. We preferred to distinguish between coding that is presented as a performance in front of an audience - live coding - and preparation for that performance, with no audience present - practice.

From this perspective, not all coding is necessarily effective practice. At one extreme, the first author's day-to-day programming requires detailed engineering work that is essential to successful performance, but might not be effective if presented as coding to an audience (e.g. the optimisation of device drivers). At the other extreme, as noted in previous research on this topic, the live coding context expects a degree of improvisation, so preparation by simply writing the same program over and over (as when playing scales on a musical instrument) seems pointless - if the program is always the same, could it not simply be retrieved from a code library? Preparation for performance should therefore involve activities that are neither original engineering, nor simple repetition. This suggests an analogy to jazz improvisation, rather than composition or classical instrumental competence.

We explored this analogy with our advisory board member Nick Cook, a professor of 'mainstream' music research, but with specialist knowledge in both performance research and digital media. Although aware of live coding as a performance genre, he had not engaged with it in the context of preparation for performance, so provided us with a fresh perspective from which to review the previous literature. This helped us to distinguish between those aspects of instrumental practice that are predetermined and intentionally over-learned (e.g. scales), those concerned with 'difficult corners' in a specific genre that might be practiced (e.g. a tricky bridge when a verse has a key change), those that develop a vocabulary of reference to other works (perhaps through riffs or 'formulae'), and those that prepare a specific 'piece' for performance (although the notes played within any given improvised performance will naturally vary).

Our current strategy has therefore been to integrate these elements into a series of daily exercises that develop fluency of low-level actions relevant to live coding. We assume that in actual performance, the program being created will be novel. But a fluent repertoire of low-level coding activities will allow the performer to approach performance at a higher level of structural abstraction - based on questions such as 'where am I going in this performance' and 'what

alternative ways are there for getting there'. In accordance with proposals by Collins and Butler, we are planning to publish a set of etudes that can develop fluency in aspects of a program related to rhythm, to harmonic/melodic structure, to texture, and to software structure. We also recognise that live coding performance is a multimedia genre, in which the contents of the screen display, and the stage presence of the performer are also essential components. A daily practice regime of preparation for performance should also incorporate etudes that develop fluency of action in these respects. These aspects of our work might be considered as constructing an 'architecture' of performance skill that complements the technical architecture of the software infrastructure.

8. CONCLUSIONS & FUTURE WORK

We have described the initial results of the Improcess project, which has developed an architecture for live coding performance motivated by the considerations of domain-specific programming language design, and by research into end-user programming. We also have an explicit concern with music performance practice that has led us to treat the integration of interface devices such as the monome as a primary architectural consideration. We have created an effective architecture with a functioning implementation, and have demonstrated that it can be used to reimplement some popular monome applications. We are now using this platform to further develop a regime of preparation for performance within a reflective research context. The Improcess environment provides a technical foundation that mirrors this musical and collaborative goal. We expect that it will enable rapid exploration of a diverse range of language options, including the potential implementation of new language features in live contexts, and even the construction of tangible performance 'instruments' that can themselves generate code processes.

We have also found that explicit reflection on interdisciplinary collaboration has been a valuable element of our research. Improcess is hosted by the Crucible network for research in interdisciplinary design, which aims to make informed contributions to public policy on the basis of projects like this [11]. In Improcess we found the intellectual contrasts between computational and musical perspectives sufficiently extreme (for example, in different team-members various interpretations of the monome as either ideally flexible or undesirably grid-like) that we represented not only multiple organisations, but multiple research cultures. We have used the Cross-Cultural Partnership template [1] to help us bring together people from these different 'cultural norms and legal frameworks for sharing culture'.

As a project demonstrating a 'performative technical practice', we believe that this juxtaposition and improvisation of cultural, technical and musical processes represents the essence of the live coding enterprise.

9. ACKNOWLEDGEMENTS

Thanks to the other members of the Improcess partnership - Tom Hall, Stuart Taylor, Chris Nash and Ian Cross - for their continued support and encouragement, and to the members of our advisory board: Simon Peyton Jones, Nick Cook, Simon Godsill, Nick Collins and Julio d'Escrivan. Thanks also to Jeff Rose and Fabian Aussems for their work on the Overtone project.

10. REFERENCES

[1] http://connected-knowledge.net/.

[2] http://docs.monome.org/doku.php?id=app:8track.

[3] http://github.com/overtone/overtone.

[4] http://github.com/samaaron/monomer.

[5] http://www.vimeo.com/290729.

[6] P. E. Agre. *Toward a Critical Technical Practice : Lessons Learned in Trying to Reform AI*. Lawrence Erlbaum Associates, 1997.

[7] P. Berg. Abstracting the Future: The Search for Musical Constructs. *Computer Music Journal*, 20(3):24–27, 1996.

[8] A. F. Blackwell. First steps in programming: a rationale for attention investment models. *Proceedings of IEEE Symposia on Human Centric Computing Languages and Environments*, pages 2–10, 2002.

[9] A. F. Blackwell, L. Church, and T. Green. The Abstract is 'an Enemy': Alternative Perspectives to Computational Thinking. *Proceedings of the 20th annual workshop of the Psychology of Programming Interest Group*, pages 34–43, 2008.

[10] A. F. Blackwell and N. Collins. The Programming Language as a Musical Instrument. *Psychology of Programming Interest Group*, pages 120–130, 2005.

[11] A. F. Blackwell and D. A. Good. *Languages of Innovation*, pages 127–138. University Press of America, 2008.

[12] A. F. Blackwell and T. Green. *Notational Systems - the Cognitive Dimensions of Notations framework*, pages 103–134. Morgan Kaufmann, 2003.

[13] J. Butler. Creating Pedagogical Etudes for Interactive Instruments. *Proceedings of the International Conferences on New Interfaces for Musical Expression*, pages 77–80, 2008.

[14] M. Duignan, J. Noble, and R. Biddle. Abstraction and Activity in Computer Mediated Music Production. *Computer Music Journal*, 34(Barr 2003):22–33, 2010.

[15] M. Fowler. *Domain-Specific Languages*. Addison-Wesley, 2011.

[16] H. Honing. Issues on the representation of time and structure in music. *Contemporary Music Review*, 9(1):221–238, 1993.

[17] E. A. Lee. Computing needs time. *Communications of the ACM*, 52(5):70–79, May 2009.

[18] H. Lieberman, F. Paterno, and V. Wulf. *End User Development*. Springer, 2006.

[19] J. McCartney. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal*, 26(4):61–68, Dec. 2002.

[20] C. Nash and A. Blackwell. Beyond Realtime Performance : Designing and Modelling the Creative User Experience. *Submission to NIME*, 2011.

[21] C. Nilson. Live coding practice. *Proceedings of the 7th international conference on New interfaces for musical expression*, page 112, 2007.

[22] A. Sorensen and A. R. Brown. aa-cell In Practice : An Approach to Musical Live Coding. *Proceedings of the International Computer Music Conference*, 2007.

[23] A. Sorensen and H. Gardner. Programming With Time Cyber-physical programming with Impromptu. *Proceedings of the ACM international conference on Object Oriented Programming Systems Languages and Applications*, pages 822–834, 2010.

[24] O. Vallis, J. Hochenbaum, and A. Kapur. A Shift Towards Iterative and Open-Source Design for Musical Interfaces. *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, (Nime):1–6, 2010.

Integra Live: a new graphical user interface for live electronic music

Jamie Bullock
Birmingham Conservatoire
Birmingham, UK
jamie.bullock@bcu.ac.uk

Daniel Beattie
Beelion Interactive
London, UK
dnl.bttie@gmail.com

Jerome Turner
User-lab, BIAD
Birmingham, UK
jerome.turner@bcu.ac.uk

ABSTRACT

In this paper we describe a new application, Integra Live, designed to address the problems associated with software usability in live electronic music. We begin by outlining the primary usability and user-experience issues relating to the predominance of graphical dataflow languages for the composition and performance of live electronics. We then discuss the specific development methodologies chosen to address these issues, and illustrate how adopting a user-centred approach has resulted in a more usable and humane interface design. The main components and workflows of the user interface are discussed, giving a rationale for key design decisions. User testing processes and results are presented. Finally, a critical evaluation application usability is given based on user-testing processes, with key findings presented for future consideration.

Keywords

software, live electronics, usability, user experience

1. INTRODUCTION

In this paper we present Integra Live, a new software application designed to address issues of software usability in live electronic music[1]. As musical practitioners working in a range of contexts including university-teaching, contemporary classical music and free improvisation, we have observed that existing software consistently presents an 'entry barrier' to musicians wishing to work with live electronics[2]. The most commonly used software for live electronics in an academic or 'contemporary classical' context is Max by Cycling 74[2][14]. Max was conceived as a 'graphical programming environment for developing real-time musical applications'[12], and as such it consists of a graphical dataflow language providing control data processing functionality for patchable digital signal processing (DSP) 'objects'. Max requires live electronics musicians to have an understanding of programming concepts such as conditional evaluation, iteration, mathematical and logical operators as well as DSP principles such as oscillation, filter design, delay buffering, table lookup and Fourier analysis. All of this is literally 'another language' to musicians who have devoted their lives to acoustic instrumental study or composition and simply want to experiment with live electronics. As a tool for dataflow programming and DSP, Max may be highly usable, but for musicians with little experience in this area, Max presents an unreasonably steep learning curve.

A number of existing projects seek to address this problem. For example, the Jamoma project[3] provdes 'a system for developing high-level modules in the Max/MSP/Jitter environment'[9], and more recently, a set of frameworks for developing Jamoma modules outside of Max[10, 11]. Jamoma offers significant advantages for both users and developers, presenting itself as a complete 'platform' within which processing modules may be used and/or developed.

An earlier project, Open Sound World (OSW), also sought to address usability issues identified in Max, by developing a new software application informed by user testing and usability evaluation[4]. However, like Max, OSW presented itself as a 'a scalable, extensible object-oriented language', and so, clearly targeted programming-savvy users rather than non-technical musicians.

Commercial software such as Bidule, Audiomulch, Reaktor, Ableton Live and Mainstage all have varying degrees of ease-of-use and applicability in live electronic music, with Bidule and Ableton Live being particularly popular with free improvisors and live electronic dance musicians respectively. However, due to its wide acceptance within academic institutions and research centres, Max remains the standard tool and entry route for composers working with electronics.

2. REQUIREMENTS

In order to verify our hypothesis that there is a need for a new application for live electronic music which is powerful yet usable for 'non-technical' musicians, we conducted a software requirements analysis. The purpose of this is to elicit requirements from stakeholders and potential users, and to analyse recorded data in order to establish design criteria.

2.1 Interviews

Four stakeholders consisting of: performer, professional composer, undergraduate composer, and post-graduate composer were interviewed in an informal setting. Interviewees were asked about their experience with existing software and informed about the aims of Integra Live and given the opportunity to respond freely about this. Some of the most salient comments are listed below.

> "I would like to see a piece of software that is more closely aligned to musical thought processes"
>
> "Current software is a big barrier for me using live electronics. It's a big deal for me to create the processing I need in my piece using Max"

[1] Music based on live processing of audio in performance
[2] `http://www.cycling74.com`
[3] `http://jamoma.org`

NIME'11, 30 May–1 June 2011, Oslo, Norway.

> "The basic processing modules should be already done so a composer can come and think about high level things. It would be nice if the most common processors were already there to be dragged or selected"

> "Max-like environments remove the element of play that you get with things like guitar pedals. These make more sense to the performer"

2.2 Online Survey

A survey of 76 potential users was conducted, drawing on Conservatoire students and staff, composers, members of new music ensembles, and members of Sonic Arts Network, Digital Music Research Network, British Computer Music and Canadian Electroacoustic Community mailing lists. 95% of those who completed the survey considered themselves to be composers with 68% considering themselves to be performers, and in general the results indicate that demographic group felt comfortable in at least two different roles. Most respondents reported that they used software for 'creation of new works', 'live performance' and 'experimentation', although 50% of respondents also use software for 'rehearsal', 'teaching' and 'writing new software components'. 78% of respondents indicated that they use 'live processing' software, with 85% indicating that they use software for 'Experimenting with sounds, controls, processing' and 76% indicating that they use software for 'performing live'.

Max was shown as being the most popular piece of software, with 21% of respondents indicating it as their favourite. SuperCollider[4] had 8% indicating it as their favourite with, Ableton Live accounting for 5%. Audiomulch and Bidule were both mentioned twice by those who indicated 'other' as their favourite software.

In addition to quantitative data gathered, the survey also recorded qualitative responses including reasons for liking or disliking specific software, and answers to the question: 'What features would you like to see in your ideal piece of music software?'. Salient responses include some of the following examples:

> "Everything. All in one. Allowing simplicity to complexity. For instance, most of people are using the basic function of Ableton Live but when you dig you can do really fancy things, programming kind of."

> "MUSICALITY. It has to work as a musical 'tool' not just as a software tool"

> Something like Max/MSP with an interface that's already in place, but that allow for infinite modification. Actually, I am thinking about AudioMulch for Mac, with a multitrack recording set up.

> An ideal piece of software would be able to be adapt to the users thought processes; this would make the software more intuitive to the user's own sense of logic. Sadly, every piece of software on the market requires a lot time just learning the program; having music software that was more 'human' would undoubtedly encourage more composers to explore different creative outlets.

> - easy way to connect any external hardware/instrument
> - few clicks to do tasks - uncomplicated first page
> - palettes of resources to choose from e.g. audio file pool, module pool

Common keywords include the following or their synonyms (number of people using the word out of the 64 who answered the section shown in brackets):

- easy (11)
- simple (4)
- like (15)
- user (9)
- interface (7)
- flexible (9)
- control (8)

The word 'like' was mostly used the context of a 'like X but with Y' idiom to indicate similarity to another piece of software e.g. '*Like* Main Stage but adapted for live electronics, simply, a piece of software like those but with flexible control over time, in concert and in rehearsal.'

2.3 ixi survey

In addition to our own surveys, conversations and interviews, we also drew on a recent survey conducted by the 'ixi' project[7] as part of our requirements gathering. This survey covers a slightly different demographic to the Integra survey, having a slightly greater emphasis towards participants with significant technical experience. This is reflected in the number of respondents reporting experience of tools that require some programming knowledge. Out of 209 survey participants, 52% indicated that they used Max/MSP, 49% indicating Pure Data and 40% indicating that SuperCollider. Interestingly, across all of the software indicated in the survey, the number of people indicating a program as their 'tool of choice' was relatively low compared to the number of users. For example, out of the 108 people that had used Max/MSP, only 35 indicated it as their tool of choice, and out of the 93 that used Reaktor, only 20 indicated it as their tool of choice. Across all of the applications in the survey the average number of users indicating a given application as their tool of choice was 17% of the total for that tool, suggesting a high level of dissatisfaction with available tools.

However, overall [7] suggests several classes of user, only some of whom are dissatisfied with available software. Particularly relevant to Integra Live are these findings:

> 'Some survey participants expressed the wish for more limited expressive software instruments, i.e. not a software that tries to do it all but "does one thing well and not one hundred things badly". They would like to see software that has an easy learning curve but incorporates deep potential for further explorations, in order not to become bored with the instrument. True to form, the people asking for such software tools had a relatively long history as instrumentalists.'[7]

2.4 Objectives

The survey results obtained, along with findings from interviews and informal conversations has led to the following observations:

1. A significant number of musicians from both acoustic and electronic music backgrounds feel that existing software for live electronics doesn't meet their requirements

[4] `http://supercollider.sourceforge.net/`

2. Many users currently use Max, so any alternative software must provide equivalent functionality, but with a musician-centred interface
3. Users require ease-of-control including the ability to easily connect external control sources
4. Users require a clean, well designed user interface that is somehow aligned with 'musical' thinking

Additionally, we aim to adhere to the following principles:

1. The software should behave like a normal application (i.e. *not* a framework or a programming environment)
2. The software should make the most common tasks easiest to achieve
3. The software should be visually appealing, and the visual design should enhance usability
4. The software should Just Work, providing low latency and stability
5. The software should be easy to download and install (good user experience)
6. The user interface should be self-explanatory [5]
7. The interface should favour standardisation over configurability
8. The software should hide complex functionality with simple UI

3. METHODOLOGY

As this project was part-funded by Integra (cf. section 8), the final application was required to be completed in a 12-month time frame. The project developers were divided between five research centres involved in the project: Birmingham Conservatoire (210 days), IEM (180 days), Notam (180 days), Malmö Academy of Music (45 days) and Muzyka Centrum (45 days). Each research centre agreed to a specific area of responsibility as follows:

- Birmingham Conservatoire: project management and core application development
- IEM: DSP module development
- Notam: Scripting functionality and Faust support
- Muzyka Centrum: module control development
- Malmö Academy: file format development and online storage

In order to deliver a user-centred application in the time frame given, a professional design company was contracted to collaborate on initial wireframes and graphic designs based on our detailed user requirements specification. This was then used as basis for an iterative development process. The Adobe AIR runtime environment was chosen as a platform for the GUI. This was due to its combination of portability, graphical richness and potential for rapid development. The Pure Data software was used as a DSP host partly for its parity with Max, but also for rapid DSP module development. Finally, the Integra Framework was developed as a middleware layer alongside the GUI to provide basic functionality such as file save/load, module management, host communications and an OSC interface[3].

Due to the disparate nature of the development team it wasn't possible to follow one specific development methodology, however, the general principles of the Agile manifesto[1] were adhered to:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Additionally, testing (user and functional) was given high priority, with testing results feeding into each iteration. This enabled us to minimise risk by finding problems early in the project.

4. APPLICATION WORKFLOW

Integra Live is divided into two main views, 'Arrange view', which is designed for arranging musical interactions in time, and 'Live view', which is designed to greatly simplify on-screen control during live performance.

4.1 Arrange View

The information architecture of the application follows a tree-like structure with a *project* at the top level. *Projects* may contain many tracks each of which may contain many blocks, each containing many modules. This is shown concisely in figure 1.

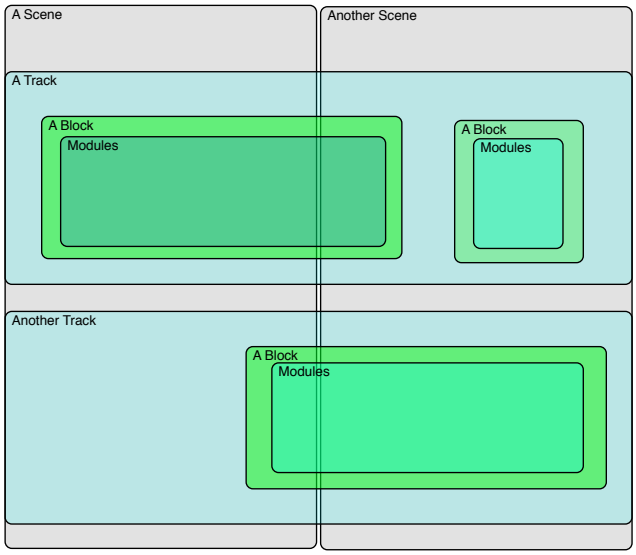


Figure 1: Integra Live information architecture

The arrange view represents this architecture visually, showing tracks as pairs of horizontal lines between which blocks can be placed. The user can navigate 'inside' blocks by clicking a '+' icon to expand them.

4.1.1 Module View

When the user navigates inside a given block, they are presented with a library of *modules* that can be dragged onto the block's canvas. *Modules* are discrete pieces of audio signal processing, synthesis or analysis functionality. It is part of the application's design that individual modules should be musically useful. That is, unlike software such as Max, which provides objects as 'building blocks' or primitives that can be combined to make more advanced units, Integra Live modules are aimed at immediate musical use. Another difference is that all Integra Live modules have pre-defined controls associated with with their attributes. These controls are displayed in the *module properties* window when a module is selected. The Integra Live core modules include a

range of filters, delays, reverbs, pitch shifter, granular synthesis, phase vocoding, resonators, soundfile playback and spatialisation. Module controls include slider, knob, range slider, x-y 'scratchpad', toggle, button as well as more specialised module-specific controls. In order to keep the user experience (UX) consistent, it was decided that the control type for each module attribute should stay fixed. That is, by design, users can't change the controls that are assigned to attributes. Figure 2 shows the module properties panel with the controls for the selected module.

Figure 2: Module properties panel containing attribute controls

Module controls (such as sliders and dials) can be used to change *Module attributes* in real-time and are interactive through clicking, dragging and text-entry. Precise values can be entered by double-clicking the control's numeric value and typing the new value in the text-entry box (figure 3).

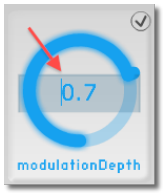


Figure 3: Entry of precise values into controls

The intended workflow in *module view* is that users create small groupings of interconnected high level modules and encapsulate these into *blocks*. The graphical presentation of modules is therefore relatively large to discourage the creation of highly complex networks of modules. If the user finds they can't achieve their goals without creating complex blocks, this would indicate a requirement for new modules providing the required functionality.

4.2 Live View

All modules and controls have a checkbox, which enables controls to be added to *Live view* either individually or per-module. The *live view* of the application shows all checked controls so that they can visualised and operated easily in live performance. Controls can additionally be resized and moved in live view. This design acknowledges that a different interaction model is required in live performance, where simplicity and immediacy of control are favoured.

4.3 Timeline

Integra Live has one master timeline, which is shown in the UI as a numbered horizontal strip near the top of the window. This provides a spatio-temporal reference point against which musical ideas can be organised. Timeline progression can be linear, where the ordering and duration of blocks corresponds to their ordering in performance, or non-linear, where the playhead moves to arbitrary points on the timeline with some blocks being activated indefinitely or stopped and started through user interaction. The playhead position can be changed manually by click-dragging the control triangle. The playhead state can be set to 'play' or 'pause' using the button controls in the top-left of the arrange view. Clicking the timeline numbering, can be used to zoom and scroll; clicking and dragging left/right scrolls, dragging up/down zooms.

Figure 4: Integra Live timeline and playhead

4.4 Envelopes

Envelopes provide a means to automate the control of module attributes over time. Envelopes are created by drawing control points into blocks in arrange view. Presenting envelopes in this way allows the user to visualise the musical 'shape' of a piece by looking at the arrangement of blocks and envelopes within the arrange view. Envelopes can be used for simple state changes (on/off) and for creating predefined musical gestures resulting from multiple module attributes changing simultaneously.

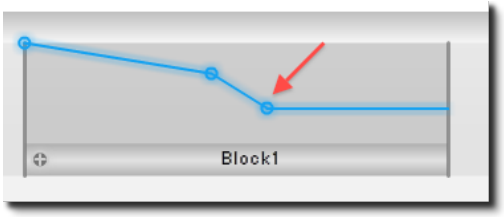


Figure 5: Adding control points to a block to create envelopes

4.5 Scenes

Scenes are used to create user-defined progressions through musical time. One application of this is to define Scenes that correspond to different sections of a musical work 'Section A', 'Section B', 'Cadenza' etc. Another application is to create multiple pathways within a work as found in improvisation and 'open form' composition. Scenes provide an additional layer on top of tracks and blocks, so that the playhead can be automatically moved to a given location on the timeline and optionally set to a 'play' state. This is achieved using *scenes*, which can be created by click-dragging in the space below the timeline. Like blocks, *scenes* have a duration, and additionally three possible states: hold, play and loop. If a scene is set to 'hold', when it is selected, the playhead will remain at the beginning of the scene. This means that all blocks under the playhead will become active and no further action will be performed unless the user gives further input to the system. If the scene is set to 'play' or 'loop' when selected, the playhead will proceed from the beginning of the scene and stop at the end of the scene or loop respectively.

4.6 Properties

When the software is in *arrange view* properties panels can be activated in the lower part of the screen by selecting entities within the UI. The properties panels follow a consistent layout, showing 'routing' and 'scripting' tabs for the

selected entity. The 'routing' tab is used to make connections between module attributes thus *routing* control data from one attribute to another. Multiple connections can be used to create one-to-many or many-to-one relations. Typical applications are for connecting external controllers such as fader boxes to DSP modules, and for 'ganging' attributes to be controlled in parallel. Activation of properties panels is shown in figure 6.

Figure 6: Block, Track and Project properties selection in arrange view

Clicking the *scripting* tab in the properties panel, provides access to the Integra Live scripting language. This is simple scripting functionality built on top of the Lua programming language. Module attribute state can be accessed from Integra script, allowing for procedural operations such as conditional evaluation and looping. The example below shows how the delayTime attribute of a TapDelay module can be modulated by side-chaining the audio input level.

```
x = AudioIn1.vu1
TapDelay1.delayTime = 100 / (x + 100)
```

5. USABILITY CONSIDERATIONS

Various features have been added across the application to improve usability and user experience. All entity instances are rename-able, allowing the user to add meaningful semantic information. For example, *scenes* can be given meaningful names like 'sectionA', 'coda', 'introduction' or 'improvsection'. Likewise, *tracks*, *blocks* and *module instances* can also be renamed.

All of these entities can also be exported from a given project and imported under a different node in the same project or into a different project. For example, the software allows a *block* to be renamed 'SpacialGranularSynth', and then exported as an Integra file for potential re-import. *Blocks* can additionally added to the in-application block library through a context menu that appears by context-clicking the *block*.

We stated in section 2.4 that Integra Live should behave like a 'normal' application. This means that it should meet the user's expectations on supported platforms, complying to the platform's human interface guidelines (HIG) as appropriate. Integra Live therefore has automatic association with its supported file type (.ixd). When a given .ixd file is double-clicked or dropped onto the application's icon, the file is opened in Integra Live as expected.

Finally, Integra Live supports infinite undo and redo for all undoable actions. Undoable actions include: adding or removing tracks, blocks, modules and scenes; changing module attributes; and renaming or moving entities.

Many of these features are standard in conventional desktop applications, but some or all of them are currently missing in commonly used frameworks and programming environments for live electronic music.

6. USER TESTING

So called 'hallway testing' [13] was employed throughout the development process particularly when significant new features were added. This was made possible by basing development at a UK Conservatoire, where potential users were easy to find and willing to offer time.

Additionally, more structured lab-based testing was conducted in the later part of the project when the software had reached a semi-stable state. User testing sessions were set-up with five users following Nielsen [8]. Each session lasted 45 minutes and consisted of:

- Introductions and coffee
- Pre-questionnaire for demographic data
- 4 structured tasks focusing on specific aspects of the software
- Post-test questionnaire gathering users' evaluation and conclusions

Tests were conducted at bespoke testing facilities provided by User-lab, part of the Birmingham Institute of Art and Design in the UK. The tests were observed by both a usability researcher and an Integra developer who was available to assist with technical problems.

An evaluation of the complete findings of the user testing process is beyond the scope of this paper, however the most salient data will be presented. The post-test questionnaire showed Max/MSP to be the most commonly used software by participants seeking to achieve similar results to Integra Live. Participants were asked to rate their experience using Integra Live using a Lickert scale [6]. The results are shown in figure 7, where the red dot indicates the average score across all five participants and the yellow bar indicates the full range of answers.

Finally participants were asked to tick words from a list they felt described their experience of using the software. Words were presented in a random order for each participant in order to eliminate potential patterns emerging as a result of word ordering. The resulting word counts were then submitted to the online word-count generator Wordle[5], with all words included in the resulting word cloud and a black font with a horizontal layout. The result is shown in figure 8.

7. CONCLUSIONS

In this paper we have described the development of Integra Live, a new software application for the composition and performance of live electronic music. We have presented our requirements gathering process, and illustrated how deficiencies in existing software can be addressed by engaging users in the development process. We have described the workflow of the new interface in detail and illustrated how the interface design is intended to meet the needs of musicians and addresses a range of usability issues. Our user testing results show that Integra Live goes part way to succeeding in its goals, but still has some way to go. In general users found Integra Live to be exciting to use, and more creative and less technical than existing software for performing similar tasks. However, whilst users found Integra

[5] `http://www.wordle.net/`

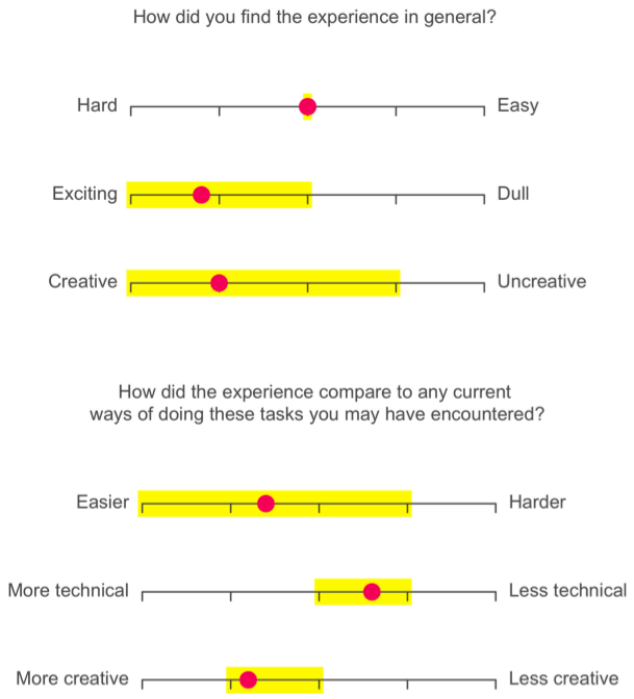


Figure 7: User testing Lickert scale results

Figure 8: Word cloud showing questionnaire word-tick response

Live 'easier' to use than existing software, they *didn't* find it 'easy'. Additionally, the software was found to be limited in terms of control-data processing functionality. Although, advanced control-data processing is possible through the scripting facility, this is clearly an unsuitable interface for entry-level use. In future work, these issues will be addressed through the addition of appropriate UI components.

8. ACKNOWLEDGMENTS

Integra Live was developed in order to meet the objectives of Integra, a 3 year EU-funded project led by Birmingham Conservatoire, following another 3-year EU-funded project, "Integra, A European Composition and Performance Environment for Sharing Live Music Technologies", both part-funded by the Culture programme of the European Commission.

9. REFERENCES

[1] K. Beck. Manifesto for agile software development. `http://agilemanifesto.org/`, Feb. 2011.

[2] J. Bullock and L. Coccioli. Towards a humane graphical user interface for live electronic music. In *Proceedings of the NIME Conference*, Pittsburgh, USA, 2009.

[3] J. Bullock and H. Frisk. The integra framework for rapid modular audio application development. In *Proceedings of the International Computer Music Conference*, Huddersfield, UK, 2011.

[4] A. Chaudhary, A. Freed, and M. Wright. An open architecture for real-time music software. In *Proceedings of the International Computer Music Conference*, Berlin, Germany, 2000.

[5] S. Krug. *DonâĂŹt Make Me Think: A Common Sense Approach to Web Usability*, chapter 1. New Rider Publishing, 2000.

[6] R. Lickert. A technique for the measurement of attitudes. *Archives of Psychology*, 50:1–55, 1991.

[7] T. Magnusson and H. M. Mendieta. The acoustic, the digital and the body: a survey on musical instruments. In *Proceedings of the NIME Conference*, New York, USA, 2007.

[8] J. Nielsen. Why you only need to test with 5 users: Alertbox. `http://www.useit.com/alertbox/20000319.html`, Feb. 2011.

[9] T. Place and T. Lossius. Jamoma: A modular standard for structuring patches in max. In *Proceedings of the International Computer Music Conference*, New Orleans, USA, 2006.

[10] T. Place, T. Lossius, and N. Peters. A flexible and dynamic c++ framework and library for digital audio signal processing. In *Proceedings of the International Computer Music Conference*, New York, USA, 2010.

[11] T. Place, T. Lossius, and N. Peters. The jamoma audio graph layer. In *Proceedings of The 13th International Conference on Digital Audio Effects*, Graz, Austria, 2010.

[12] M. Puckette. Combining event and signal processing in the max graphical programming environment. *Computer Music Journal*, 15(3):68–77, 1991.

[13] J. Spolsky. *User Interface Design for Programmers*, chapter 13. Springer, 2001.

[14] D. Zicarelli. An extensible real-time signal processing environment for max. In *Proceedings of the International Computer Music Conference*, San Francisco, USA, 1998.

Robust and Reliable Fabric, Piezoresistive Multitouch Sensing Surfaces for Musical Controllers

Jung-Sim Roh
Fashion Textile Center
Seoul National University
Korea
simi1012@snu.ac.kr

Yotam Mann
CNMAT
Dept. of Music
UC Berkeley
yotammann@berkeley.edu

Adrian Freed
CNMAT
1750 Arch Street
Berkeley, CA 94709
adrian@cnmat.berkeley.edu

David Wessel
CNMAT
Dept. of Music
UC Berkeley
wessel@cnmat.berkeley.edu

ABSTRACT
The design space of fabric multitouch surface interaction is explored with emphasis on novel materials and construction techniques aimed towards reliable, repairable pressure sensing surfaces for musical applications.

Keywords
Multitouch, surface interaction, piezoresistive, fabric sensor, e-textiles, tangible computing, drum controller

1. INTRODUCTION
Multitouch array sensing with flexible substrates has been experimented with in the last three decades primarily for robotics and medical sensing applications [28]. Most of the research has been on core sensing and materials questions. The novel contributions reported here primarily involve the integration challenges of flexible surface sensing for musical controllers where reliability, repairability and robustness have to be addressed in addition to the sensing and materials challenges. These less glamorous issues have been mostly ignored in academic and hobbyist music controller designs that are rewarded more for apparent "novelty" and potential than long-term viability. We suggest that the "New" in NIME also refers to the experience for the performer that their controllers perform "like new" every time they play them–for decades to come.

We focus here on piezoresistive, multitouch sensing surfaces because the popular capacitive multitouch systems [15] do not provide sufficient taxel pressure resolution for musical applications. CNMAT's work for guitars [40] with Tactex Controls Inc. on the Kinotex [18] optical cavity sensing approach proved too difficult to scale to the high taxel counts required to capture nuanced musical gestures over large surfaces. FTIR optical surfaces have been developed with pressure sensitivity but they require bulky camera systems and have limited sensing bandwidth due to the slow frame rate of the cameras [32]. Systems fusing muscle sensing with surface sensing show some promise but the long reported latencies [2] (150ms) preclude them from most musical applications.

2. History of Piezoresistivity
Piezoresistivity, the modulation of electrical resistance according to stress, is observed in elemental materials, semiconductors and composites. This property has been exploited in sound and musical applications for well over a hundred years although it wasn't necessarily understood as such in earlier times [33].

A significant application of piezoresistivity in audio is Alexander Bell's microphone using carbon rods under strain. The subsequent commercial success of the telephone resulted in numerous refinements of carbon-based piezoresistive microphones from Edison, Berliner, Blake, Hughes, Hunnings and White and others. By the late 1800's piezoresistive microphones and liquid/conductor current modulation were broadly understood by engineers and were techniques used routinely in early electronic musical instruments.

Singer's 1893 patent [31] is important because its single claim is a carbon-based piezoresistive sensor for keyboard musical instruments. This patent signals the use of carbon granules in an elastomeric substrate (rubber in this case) a technique still widely-employed and for which patents are still regularly issued, e.g., US6820502 [34] (with no less than 80 claims).

The application Singer proposes is also congruent to established musical interaction design patterns [38], i.e. where sound dynamics are controlled by surface pressure gestures.

Piezoresistive pressure sensing techniques can be found in new electronic musical instruments and controllers throughout the twentieth century, and to the present day where they appear commercially at the rate of several products each year. There isn't enough space here to survey them all properly –a search in European patent listings resulted in over 50 entries for musical instruments employing resistive sensing. We should however mention that the expressiveness valued in the early instruments of the field, the Ondes Martenot [17], Heliphon [11] and Trautonium [35, 36] is attributable to their use of resistive pressure sensing to control sound dynamics.

In 1982 Franklin Eventoff introduced Force Sensing Resistors (FSR); printed piezoresistive sensor assemblies designed with musical applications in mind [7]. His firm developed low-cost, high volume manufacturing techniques for printing conductive ink and piezo-resistive polymer matrices onto plastic substrates. These devices are routinely used in NIME projects [16]. The advantage of using manufactured sensors is that they can be replaced from readily available spares instead of being repaired. This same may be said of steel and nylon guitar strings, for example.

When the Ondes Martenot touch key no longer performs as the player wishes they have to remix a new batch of a "magic" piezoresistive concoction (carbon and mica) that is placed in a leather sensing bag under the key [10]. This is an example of how the design of musical instruments assemblies involves deciding whether they are to be adjusted, repaired or replaced by the performer, a "technician", a luthier or at a factory. The concept of Mean Time To Repair (MTTR) is useful here. A common design pattern is to place the repair, tuning and replacement of the most fragile or stressed components in the hands of the performer. The oboe is an interesting case because

NIME'11, 30 May–1 June 2011, Oslo, Norway.

of the high skill level required to construct a reed that is discarded–often after only a few hours of use. The Ondes Martenot touch key does not have to be replaced often but now there is such a small network of technicians that the replacement task defers to performers.

This paper will focus on design choices for piezoresistive multitouch surfaces driven by considerations of robustness and repairability and consistency of performance with the intention of guiding the development of instruments that can be played and maintained for the lifetime of the performer and beyond.

3. Materials

The core sensors we introduce are built from piezoresistive non-woven fabric from Eeonyx [8] and a new conductive, composite thread spun from silver-plated copper wire and polyester yarn [20-23].

3.1 Piezoresistive Substrate

Conductive-polymer-infused, non-woven fabrics have been refined by Eeonyx to minimize hysteresis and provide uniform resistivity with long-term stability. We rejected the option of conductor-loaded elastomeric materials (such as Zoflex [19]) because they exhibit higher hysteresis than fabric and paper and the conductor-loading considerably weakens the material–a concern in high impact situations such as drumming. Carbon-loaded paper has been proposed for these applications [14] with the idea that it so cheap that it can be simply discarded and replaced as needed. This may be true for a single point pressure sensor but it isn't for large-taxel-count multitouch where connection and integration costs far outweigh material costs.

Although there are indications that paper sensors give comparable performance to fabric, such experiments were done in laboratory conditions with fresh materials. An advantage of fabrics is that they can be engineered from a wide variety of base materials to last longer and be less susceptible to environmental and insect damage than paper.

Fabrics can be stretched and compressed making it possible to fit them to a broader class of surfaces [37] than paper which is limited to developable surfaces (Guassian curvature 0).

3.2 Robust, Solderable Conductive Thread

Piezoresistive materials may be sensed either by sandwiching them between two conductors or laying them onto a interdigitated grids of conductors. The designs proposed here use the former approach rather than the latter because it scales well to large taxel counts. This is because conductors can be simply arranged in parallel lines in orthogonal directions on each side of a piezoresistive patch of fabric.

Numerous choices for conductors in this application have been explored. We have chosen to use conductive embroidery thread because unlike printed silver inks, for example, threads can be easily repaired with readily-available tools and moderate skill level. With fabric and thread the core geometry of the instrument can be determined by the performer – just as the reed is customized by each oboe player.

This customizability is also valued in the wearable electronics field. Although suitable for quick exploratory prototypes, we have learned we cannot directly adopt the readily available materials and components from the hobbyist e-textile and wearables community such as the Arduino Lilypad or silver-plated nylon thread: the resulting assemblies are not robust enough for extended use and the gesture signal processing performance and transmission is not fast enough.

The spun-metal thread we are using has the advantage over silver-plated nylon thread of being less influenced by the effects of corrosion because of a higher overall metal content of the silver-plated copper wire. Another important advantage is that with an appropriate layout these threads may be soldered. This greatly eases the challenge of reliably connecting the sensor array with conventional electronic circuits. With plated threads we have observed an alarming increase over time of the electrical resistance of connections–something that does not occur with a spun metal thread.

4. Implementation Topography

We have explored four different ways of implementing the well-known design pattern for flexible-surface, resistive multitouch [5]. We have confirmed that the different implementations performed similarly as position and pressure sensing arrays. This is not surprising since the basic piezoresistive material, conductive threads and spacing were the same for all of them. However laboratory-condition sensor performance is just one of many important properties we need to evaluate for these implementations. We will discuss the build, repair and maintenance issues of each.

4.1 Sandwich

In this implementation the piezoresistive fabric is sandwiched between two non-conductive pieces of fabric each of which has parallel lines of conductive thread sewn into them. The three component fabrics are held together with a sparse array of light tension stitches. The outside fabric pieces are of course arranged so the threads on top and bottom are orthogonal to each other.

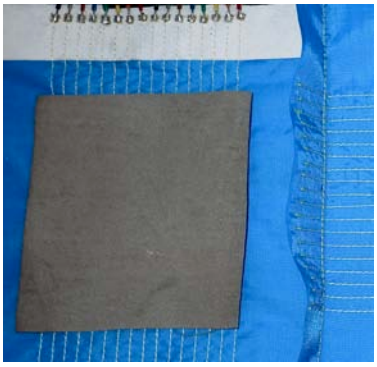


Figure 1: Sandwich

4.2 Machine Sewn

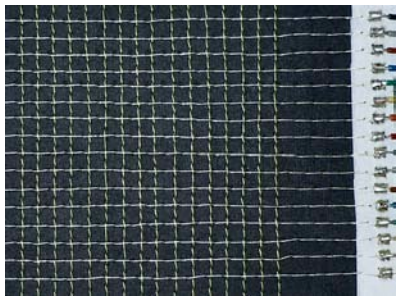


Figure 2: Machine Sewn

An ordinary sewing machine is set up with a conductive bottom thread and insulating top thread. As parallel lines are sewn into the piezoresistive fabric thread tensions are carefully adjusted (or "misadjusted" according to conventional sewing norms) to minimize the possibility of conductive thread being pulled on to the wrong side of the fabric. After one side is complete the fabric is turned over and an orthogonal array of lines is sewn in.

Shorts between the layers can be avoided by careful positioning of the stitches on the second side.

4.3 Trapped Conductor

An ordinary sewing machine is used to lay in narrow rows and columns of insulating zig-zag couching stitches on respective sides of the piezoresistive fabric. Conductive threads are then interlaced through the couching stiches by hand. Note that a variety of couching stitch styles could be employed including straight stitches or wide patterns. We favor the zig-zag stitch because interlacing the conductive threads is facilitated by the straight line path through them.

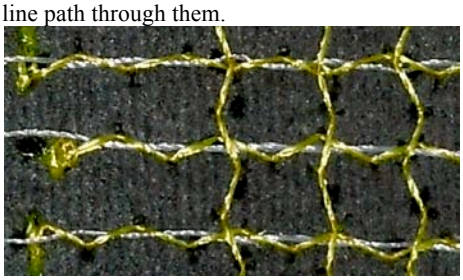


Figure 3: Trapped Conductor

4.4 Woven

The woven implementation [29, 30] is the only one where the conductive threads periodically alternate sides of the piezoresistive fabric. The periods of these running stitches are synchronized so that orthogonal runs are always on opposite sides of the fabric.

These runs are created by hand, a process that can be sped up somewhat by punching or cutting an array of holes in the substrate fabric.

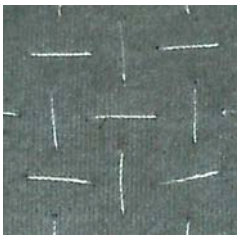


Figure 4: Woven

4.5 Discussion and Comparison of Construction Methods

The sandwich construction offers advantages for large production volumes because the outer layers can be cut from long, wide rolls of pre-manufactured fabric.

Figure 5: Strain-relieved cabling and solder pad

We illustrate this in Figure 5 using a woven fabric that integrates stripes of conductive yarn made by Eleksen.

Note the use of spun-metal thread to transition between the conductive yarn and soldered connections to a flat cable. The sandwich construction also allows a worn or damaged piezoresistive patch to be easily replaced independently of the conductive components.

The machine-sewn construction is accessible to hobbyists on introductory-model sewing machines and results in a thin, one-piece assembly. Repairs to broken or frayed threads may be done by hand with the appropriate equipment.

The trapped conductor approach is the fastest to repair because only a needle and thread are required to replace an entire row or column.

The woven approach lends itself to rapid hand repair but the regular switching from one side of the substrate fabric to the other is not easy to automate. This means this approach will be rather labor intensive for high taxel counts. However it is worth considering when an application demands hand construction for other reasons–for example, when an array is required to match a non-rectangular and non-convex shape with cutouts such as the top plate of a guitar. Hand construction allows the density of sensory node points to be modulated throughout the surface and the integration of insulating patches to route conductors around holes and concavities.

5. Sensor Data Acquisition and Gesture Analysis

An important axis in the design space of touch sensing devices represents a trade-off between complexity of the sensing surface and complexity in the data acquisition hardware and software systems. Capacitive multitouch [39] is popular in high volume consumer products because the complexity of the sensing challenge can be concentrated in a single integrated circuit that performs analog and digital signal processing, calibration, gesture interpretation, data formatting and transmission. Capacitive sensor arrays are built at low cost per taxel by etching or printing processes that are already well-established for multiplexed, flat displays. The difficulty of this particular point in the design space for musical instrument applications is that only standardized sizes and shapes are available and pressure sensing performance is still poor or completely unavailable.

The 4-wire XYZ pad [13, 26, 42] shows that good pressure and position performance can be achieved at a range of sizes using fabric. These can be assembled in ten minutes with ordinary tools and skill. The basic data acquisition algorithms required are also accessible to hobbyists. However the spatial precision of this device depends on the uniformity of the piezoresistive fabric and uncoupling of the interaction between pressure and position. This can be achieved using algorithms less accessible to hobbyists [27]. The touch surfaces explored in this paper address these difficulties by moving some of the complexity into the construction, increasing the number of conductors across the surface so that the spatial resolution is mostly determined by the positioning accuracy of the conductive threads–not the uniformity of the piezoresistive substrate.

If gestures are assumed to be constrained to single touches in predetermined regions the gesture interpretation software is of moderate complexity and accessible to non-specialists. The full potential of high density arrays of pressure sensors is realized when force profiles of large numbers of objects placed anywhere on the surface can be analyzed and for which sophisticated machine vision algorithms are usually employed. This complexity can be managed by partitioning the design so that the electronics and software integrated into the touch surface delivers an uninterpreted taxel image that can be

processed by a target computer system to the desired level of detail.

5.1 Multiplexed scanning of the surface

Multiplexing is the approach available for moderate-to-large taxel counts. The cost of connections is too high in these applications for each pressure sensing point in the array to be separately wired to a data acquisition channel. The main technical challenge with multiplexing resistive arrays is to isolate the resistance change at a particular node in the array from neighboring resistance changes. Many approaches to this are known [12] and some have been evaluated in terms of acquisition performance and implementation complexity [6]. These older complexity measures (that count discrete components such as op-amps, drivers, analog switches etc.) are not very useful in current designs because these components are now integrated into embedded microcontrollers. Also component costs have dropped to a point where integration and connection costs now dominate designs. For this reason there is a renewed interest in data acquisition techniques that involve wiring the sensing array directly to the microcontroller with a minimum number of external components.

By writing software that dynamically changes the function of microcontroller pins from outputs to A/D conversion inputs we have implemented data acquisition without any other external components for both XYZ pads and multitouch arrays. However in the multitouch case it is difficult to eliminate lateral current flows across the piezoresistive fabric without additional electronics to avoid cross-talk contaminating the taxel pressure estimates.

Developers of the "UnMousePad" [24] introduced anisotropicity in the conductivity of the piezoresistive materia (by adding conductors) to reduce crosstalk [25]. Their demonstrations were built with established conductive ink printing processes where these additional conductors add no cost. Since our sensors are embroidered their approach would result in an increase cost in time and materials on the surface. We have found that a careful, efficient implementation of current nulling approaches [9] addresses crosstalk without substantially increasing systems costs. A single op-amp and resistor are required for each of either the row or column conductors. An important advantage of the current nulling approach is that an entire row or column of taxels can be measured concurrently–an essential requirement to achieve a high taxel frame rate (8kHz) for large arrays.

6. Integration Design and Implementation

It is a formidible challenge to reliably connect the conductors from a flexible sensing surface that is subject to high continuous and impulsive stress and strain to the relatively rigid and unyielding circuit boards holding the sensor acquisition electronics. Promising techniques are being developed for intrinsic fabric electronics and flexible circuit constructions for extrinsic electronics for textile applications. A recent adaptation of chip-on-board techniques to a fabric substrate is attractive [41]. However low-volume or hobbyist application of these approaches is unlikely in the near future especially when high connection counts are required.

We summarize here a series of exploratory experiments that have yielded viable solutions to the connection problem for low production and hobbyist applications.

The following table quantifies the problem. It describes variations in electrical contact resistance of various approaches to connecting conductive spun-wire thread to circuit boards including wrapping, gluing with conductive epoxy and soldering.

Further work is needed in this area to develop stronger data with analysis of long-term failure processes with accelerated life testing etc. but we have enough data to conclude that the particular contact method matters.

Table 1 Connection resistance

	Before Mechanical Stress		*After Mechanical Stress*	
Connection	avg resistance (ohms)	std dev	Avg. resistance (ohms)	std dev
10x wrap	.43	.16	.25	.034
5x wrap	.202	.053	.271	0.04
5x wrap + CircuitWork conductive epoxy	.255	.036	fail	n/a
5x wrap + MG chemicals conductive epoxy	.189	.015	.149	.023
5x wrap + solder on same hole	.142	.0144	.17	.024
0x wrap with stress relief on adjacent hole	.263	0.05	fail	n/a

We found that the fastest method for hand construction with readily-available tools is soldering. In automated production conductive glues are attractive because they can be printed, and achieve higher densities than soldering because they don't require special structures to dissipate heat. We focus now on soldering because construction is rapid, repair is much faster than conductive glues. Solder also has a longer shelf-life and easier accessibility than conductive glues.

7. Soldered textile connections

7.1 Strain relief wrapping

Since a soldered joint holds the thin wires of the spun-metal thread to a large rigid surface the wires can quickly suffer metal fatigue and breakage unless strain relief is provided. One solution illustrated here is to lace the thread into a pair of holes and then solder the metal plies to a third contact point on the board. We prefer this approach to the scheme on the Arduino Lilypad [3, 4] of wrapping the thread between a hole and the sharp edge of a fiber-glass circuit board. The edge of these boards is much more abrasive than a plated hole. In any case it is essential that the circuit board be well anchored to the substrate fabric with other threads specifically optimized for this purpose so that the stress of snags and pulls of the board are absorbed by them instead of the conductive threads. To better control these two different roles of threads it is advantageous to provide a zig zag and loose path for the conductive threads.

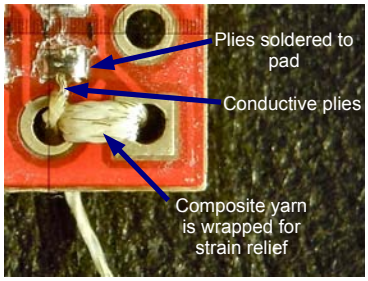


Figure 6: strain relief and soldered thread

Another alternative for circuit board connections is to use existing multiwire cabling assemblies such as ribbon cables and solder these to specially constructed fabric solder pads as

shown in Figure 5 and described more fully in the next section. Notice that the flat cable is anchored to the substrate fabric with insulated-thread stitches in the interstices of the conductors.

Figure 7 shows the compatibility of the spun-metal thread with flexible circuit board materials, another approach to strain relief:

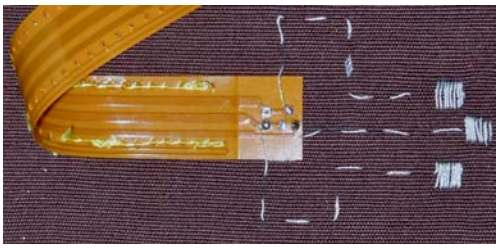


Figure 7: Sewing to flat cable

7.2 Embroidered Soldering Pad and Via

Solder melting points are higher than the melting and combustion temperatures of many fabric materials. Our solution with spun-metal embroidary thread is to sew pads on cotton tape that serves to insulate the polyester piezorestive fabric and to provide a large enough pad area for the heat to be rapidly dissipated across its surface.

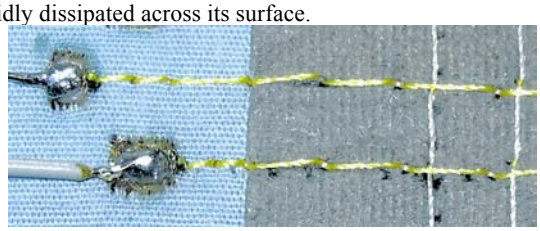


Figure 8 Soldered thread

Note that by embroidering the pads with conductive thread on both sides a via is formed allowing for all the soldered connections for the array to be implemented on one side of the fabric.

The space is left in the layout design for sewing a new pad to connect to in the case of failures. This is more reliable ultimately than reusing the original pad.

In the example of Figure 5 various circuit paths are explored. Alternating access to each side of a run increases the density of the pads. Connecting both sides of a run provides a redundant path that allows a run to still function correctly with a single breakage anywhere along it.

8. Software considerations

Before driving current into the array a special sequential scan is done to identify shorted and broken conductors. Shorts across the two layers require immediate repair. Broken connections and shorts between adjacent conductors can be compensated for in the software.

During regular use any unusual pressure data triggers the same power-on evaluation sequence. The idea is to provide enough resiliance so that a performer need not abort a performance when a single wire breaks or shorts.

To minimize performance bottlenecks from USB we encode an array scan as compact OSC blobs with one byte per taxel.

9. Performance and Spatial Density

Readily available 8-bit microcontrollers with integrated USB can scan a 12x12 taxel array at around 200Hz. The limiting factor in these chips is the A/D conversion speed. 32-bit microconrollers with 16 channels of A/D conversion are available with higher A/D conversion rates and can scan a 16x16 taxel array at 500Hz. At this point the USB implementations become the limiting factor. This can be addressed by tiling 4 multitouch arrays and microcontrollers and aggregating the four streams with a USB 2.0 hub. For arrays larger than 32x32 custom hardware using FPGA's is a better way of managing the necessary parallelism to achieve high sample rates and host computer communication rates [38]. This is the scale where hobbyist fabric multitouch becomes challenging. The impact of these constraints is application dependent because conductor and spatial sampling density are still free variables of the design. We have chosen to explore 2.5mm and 5mm spacing. We believe the former may provide sufficient spatial resolution to estimate both finger position and orientation on the surface. The latter is sufficient to estimate finger position with sufficient resolution to capture vibrato-like gestures quite well. In this recording below the vertical axis represents displacements in mm of the second nuckle of the performers index finger. This measuring point is 5.4cm from the tip of the finger at the sensing surface.

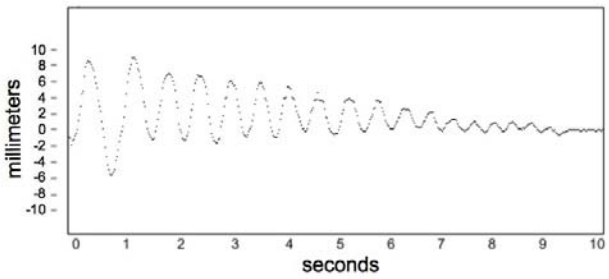


Figure 9. Vibrato Gesture pressure sensing

10. Conclusion and Future Work

We have achieved 7-bit dynamic range of pressure sensing at each taxel of our piezoresistive multitouch arrays. The piezoresistive fabric we are using was designed for foot pressure measurements so we have excellent sensitivity for ballistic interactions and high pressures. We will refine the materials to improve this sensitivity for light, stroking gestures.

As we gain more experience with our smaller arrays we will scale up and move away from USB 2.0 to USB 3.0 or Gibabit Ethernet to approach our goal of an 8kHz taxel frame rate. At these rates multicore parallel computation will be required to execute the machine vision algorithms for effective taxel scene analysis [1].

We have demonstrated new design techniques to create robust, reliable and repairable multitouch. We continue to explore faster and easier methods of construction both for small scale manufacturing and individual hobbyists.

11. Acknowledgements

We gratefully acknowledge the support of Meyer Sound Labs, Disney/Pixar and Concordia University, Faculty of Fine Arts.

12. Bibliography

[1] Battenberg, E., Freed, A. and Wessel, D. Advances in the Parallelization of Music and Audio Applications *ICMC*, CMA, 2009.

[2] Benko, H., Saponas, T.S., Morris, D. and Tan, D. Enhancing input on and above the interactive surface with muscle sensing *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ACM, Banff, Alberta, Canada, 2009, 93-100.

[3] Buechley, L., Eisenberg, M., Catchen, J. and Crockett, A. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education *Proceeding of the*

twenty-sixth annual SIGCHI conference on Human factors in computing systems, ACM, Florence, Italy, 2008, 423-432.

[4] Buechley, L. and Hill, B.M. LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, ACM, Aarhus, Denmark, 2010, 199-207.

[5] Burgess, L.E. Tactile Sensing Transducer. USPTO #5060527, 1991.

[6] D'Alessio, T. Measurement errors in the scanning of piezoresistive sensors arrays. *Sensors and Actuators A: Physical*, *72* (1). 71-76, 1999.

[7] Eventoff, F.N. Electronic Pressure Sensitive Transducer Apparatus. USPTO #4314227, 1982.

[8] Freed, A. Application of new Fiber and Malleable Materials for Agile Development of Augmented Instruments and Controllers *NIME 2008*, 2008.

[9] Freed, A. Novel and Forgotten Current-steering Techniques for Resistive Multitouch, Duotouch, and Polytouch Position Sensing with Pressure *NIME 2009*, 2009.

[10] Guillot, I., Hartman-Claverie, V. and Vaiedelich, S. Maurice Martenot: La poudre de l'enchanteur *CIM09*, Paris, France, 2009.

[11] Helberger, B. Electrical Musical Instrument. USPTO #2201232, 1938.

[12] Hillis, W.D. A High-Resolution Imaging Touch Sensor. *The International Journal of Robotics Research*, *1* (2). 33, 1982.

[13] Kazuhiko Ito, F.I.U., Myagi. Input Device. USPTO #4529959, 1985.

[14] Koehly, R. Fabrication of Sustainable Resistive-Based Paper Touch Sensors: Application to Music Technology (forthcoming). Doctorate Thesis McGill University 2011.

[15] Lee, S., Buxton, W. and Smith, K.C. A multi-touch three dimensional touch-sensitive tablet. *SIGCHI Bull.*, *16* (4). 21-25, 1985.

[16] Marshall, M. and Wanderley, M. Evaluation of sensors as input devices for computer music interfaces. *Computer Music Modeling and Retrieval*. 130-139, 2006.

[17] Martenot, M. Résistance variable pour instrument de musique électriques et radio-électriques. Industrielle, D.d.l.P. #703923, 1930.

[18] Reimer, E. and Baldwin, L. Cavity Sensor Technology for Low Cost Automotive Safety and Control Devices *Air Bag Technology 99*, Detroit, 1999.

[19] RFMicroLink Zoflex. 2011. http://www.rfmicrolink.com/products.html.

[20] Roh, J., Chi, Y. and Kang, T. Thermal insulation properties of multifunctional metal composite fabrics. *Smart Materials and Structures*, *18*. 025018, 2009.

[21] Roh, J., Chi, Y. and Kang, T. Wearable textile antennas. *International Journal of Fashion Design, Technology and Education*, *3* (3). 135-153, 2010.

[22] Roh, J., Chi, Y., Lee, J., Nam, S. and Kang, T. Characterization of embroidered inductors. *Smart Materials and Structures*, *19*. 115020, 2010.

[23] Roh, J., Chi, Y., Lee, J., Tak, Y., Nam, S. and Kang, T. Embroidered Wearable Multiresonant Folded Dipole Antenna for FM Reception. *IEEE Antennas and Wireless Propagation Letters*, *9*. 803, 2010.

[24] Rosenberg, I. and Perlin, K. The UnMousePad: an interpolating multi-touch force-sensing input pad. *ACM Trans. Graph.*, *28* (3). 1-9, 2009.

[25] Sado, R. Anisotropically Pressure-Sensitive Electroconductive Composite Sheets and Method for the Preparation Thereof. USPTO 1981.

[26] Sandbach, D.L. Multiplexing Detector Constructed from Fabric#US6492980, 2001.

[27] Schmeder, A. and Freed, A. Support Vector Machine Learning for Gesture Signal Estimation with a Piezo Resistive Fabric Touch Surface *NIME*, Sydney, Australia, 2010.

[28] Scilingo, E.P., Lorussi, F., Mazzoldi, A. and De Rossi, D. Strain-sensing fabrics for wearable kinaesthetic-like systems. *Sensors Journal, IEEE*, *3* (4). 460-467, 2003.

[29] Shimojo, M., Makino, R., Namiki, A., Ishikawa, M., Suzuki, T. and Mabuchi, K., A sheet type tactile sensor using pressure conductive rubber with electrical-wires stitches method. in, (2002), IEEE, 1637-1642.

[30] Shimojo, M., Namiki, A., Ishikawa, M., Makino, R. and Mabuchi, K. A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method. *Sensors Journal, IEEE*, *4* (5). 589-596, 2004.

[31] Singer, P. Electric Musical Instrument. USPTO #501543, 1893.

[32] Smith, J.D., Graham, T.C.N., Holman, D. and Borchers, J., Low-Cost Malleable Surfaces with Multi-Touch Pressure Sensitivity. in *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on*, (2007), 205-208.

[33] Strangways, H. The Action of the Telephone. *THE TELEGRAPHIC JOURNAL AND ELECTRICAL REVIEW*. 216, 1882.

[34] Tongbi Jian, Z.W. High Resolution Pressure-sensing device having an insulating flexible matrix loaded with filler particles. USPTO #6820502, 2004.

[35] Trautwein, F. Device for the Production of Musical Sounds by Electrical methods. UK #403365, 1933.

[36] Trautwein, F. Electrical Musical Instrument. USPTO #2141231, 1938.

[37] Wang, C.C.L., Tang, K. and Yeung, B.M.L. Freeform surface flattening based on fitting a woven mesh model. *Computer-Aided Design*, *37* (8). 799-814, 2005.

[38] Wessel, D., Avizienis, R., Freed, A. and Wright, M., A force sensitive multi-touch array supporting multiple 2-D musical control structures. in *NIME*, (2007), ACM, 41-45.

[39] Westerman, W. Hand Tracking, Finger Identification, and Chordic Manipulation on a Multi-touch Surface. Ph. D. Thesis Ph. D.University of Delaware 1999.

[40] Wright, M., Freed, A., Lee, A., Madden, T. and Momeni, A. Managing Complexity with Explicit Mapping of Gestures to Sound Control with OSC *International Computer Music Conference*, International Computer Music Association, Habana, Cuba, 2001, 314-317.

[41] Yoo, J., Yan, L., Lee, S., Kim, H. and Yoo, H. A wearable ECG acquisition system with compact planar-fashionable circuit board-based shirt. *Information Technology in Biomedicine, IEEE Transactions on*, *13* (6). 897-902, 2009.

[42] Yoshikawa, O. Pressure Sensitive Three-Dimensional Tablet and Manipulation Data Detecting Method Therefor. USPTO #5852260, 1998.

Examining the Effects of Embedded Vibrotactile Feedback on the Feel of a Digital Musical Instrument

Mark T. Marshall
Interaction and Graphics Group
Department of Computer Science
University of Bristol, UK
mark@cs.bris.ac.uk

Marcelo M. Wanderley
Input Devices and Musical Interaction Laboratory
Music Technology Area
McGill University, Canada
mwanderley@music.mcgill.ca

ABSTRACT

This paper deals with the effects of integrated vibrotactile feedback on the "feel" of a digital musical instrument (DMI). Building on previous work developing a DMI with integrated vibrotactile feedback actuators, we discuss how to produce instrument-like vibrations, compare these simulated vibrations with those produced by an acoustic instrument and examine how the integration of this feedback effects performer ratings of the instrument. We found that integrated vibrotactile feedback resulted in an increase in performer engagement with the instrument, but resulted in a reduction in the perceived control of the instrument. We discuss these results and their implications for the design of new digital musical instruments.

Keywords

Vibrotactile Feedback, Digital Musical Instruments, Feel, Loudspeakers

1. INTRODUCTION

Most traditional musical instruments inherently convey an element of tactile feedback to the performer in addition to their auditory and visual feedback. Reed instruments produce vibrations which are felt in the performer's mouth, string instruments vibrations are felt through the fingers on the strings, or through contact between the performer's body and the resonating body of the instrument [4]. This tactile feedback leads to a tight performer-instrument relationship which is not often found in digital musical instruments.

Studies have shown that while beginners make extensive use of the visual feedback provided by musical instruments, in expert performance it is the tactile and kinaesthetic which is the most important [7]. The majority of digital musical instruments provide only auditory and visual feedback to the performer, which results in a less complete sense of the instrument's response to the player's gestures than is available with traditional instruments [4]. It has also been stated that only the physical feedback from an instrument is fast enough to allow a performer to successfully control articulation [11].

In a previous work [9], we presented a digital musical instrument that uses embedded loudspeakers to produce vibrotactile feedback that is directly based on the sound being created by the instrument. In this paper we examine in more detail the ways in which this feedback can be created, compare the vibrations produced with those of an acoustic instrument and examine how the addition of this feedback affects the performer's perception of the "feel" of the instrument.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. PRODUCING INSTRUMENT-LIKE VIBRATIONS

One possible use of a vibrotactile feedback system in a digital musical instrument is to produce vibrations that are based on the sound the instrument is producing. In an acoustic instrument the sound production mechanism also produces the vibrations that the performer feels. If we wish to provide vibrations in a DMI that are produced in a similar way to those of an acoustic instrument, these vibrations must then be directly linked to the sound production. Such a link can be achieved by deriving the vibrotactile feedback signal from the sound synthesis output of a DMI.

In order to physically produce these vibrations then, an actuator is needed which meets the following requirements:

1. Capable of producing the full frequency range of human tactile sensation.
2. Offer independent control of frequency, amplitude and waveform.
3. Offer a large range of amplitude control (to allow for instrument dynamics).
4. Be driven by an audio signal, or a signal easily derived from an audio signal.

As discussed in [9], we can see that voice-coil, the tactor and the piezoelectric element each meet these requirements to different extents. Of these, the voicecoil offers the greatest range of frequency and amplitude control. Also of interest is that if we use a voicecoil in the form of a loudspeaker, then the system can also be used as the main sound production method of the instrument. This not only adds sound-related vibrotactile feedback to the instrument but also co-locates the sound production into the instrument itself [5, 1].

2.1 Vibrotactile Feedback from the Sound Synthesis System

By routing the sound output from the sound synthesis system in a DMI to the an amplifier and loudspeakers within the instrument body we can produce instrument-like vibrations within the DMI itself. This was the approach that we

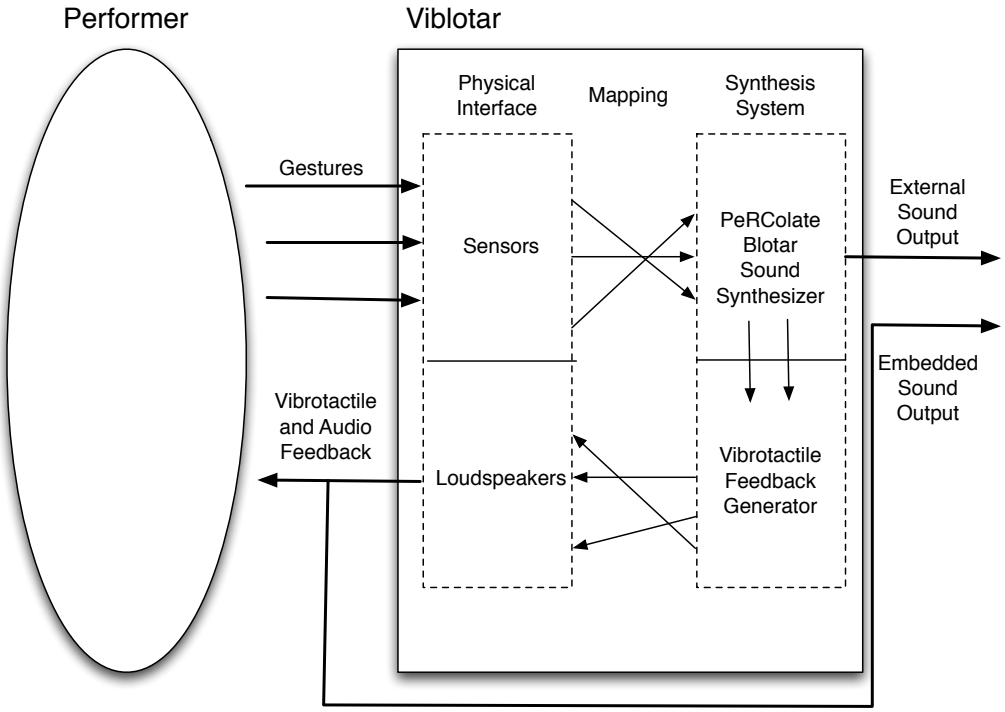


Figure 1: The overall structure of the Viblotar [9], based on the model of a DMI presented in [8].

took in the development of the Viblotar [9]. This section details the components of the Viblotar and the methods by which it generates vibrotactile feedback for the performer.

Figure 1 gives an overview of the components of the Viblotar. The output of the sound synthesis system is used to drive both the *external* sound production and the vibrotactile feedback (and *internal* sound production) components. The internal sound production mechanism consists of the amplifier and loudspeakers embedded in the instrument body. This also acts as the vibrotactile feedback component as the loudspeaker output also creates vibrations in the instrument body. The external sound production would be any amplifiers or external loudspeakers, which could be used to provide amplified sound for performance in a larger space. In many cases the internal and external sound production would be driven using the same signal, so that the external sound is an amplified version of the internal sound. However, the use of separate internal and external sound production mechanisms allows for some interesting effects which will be discussed in more detail in the next section.

In the Viblotar the output from the sound synthesizer is fed to the input of a frequency response modification system. which uses parametric equalizer sections to modify the signal to change the frequency response of the Viblotar output. This modified signal is then sent through a digital to analog converter (DAC), the output of which is a line level audio signal which is fed to the hardware of the Viblotar's vibrotactile feedback component. There it is amplified and output through the embedded loudspeakers.

When the sound synthesis signal is fed directly through the vibrotactile feedback generator, without any modification of the signal, then the vibrotactile feedback provided by the Viblotar is directly related to the sound of the instrument. The sound produced by the embedded loudspeakers is the sound of the instrument itself and this sound causes vibrations in the instrument. However, it is also possible to modify the signal used to drive the vibrotactile feedback component. In this case, the vibrotactile feedback would still be related to the sound produced by the instrument, without being directly caused by it. By using the unmodified signal to drive the external sound production and a modified signal to drive the vibrotactile feedback and internal sound production we can create a number of interesting feedback effects.

2.2 Modifying the Vibration Response

The availability of both internal and external sound production mechanisms in the Viblotar allows 3 main modes of operation:

Internal sound production only: in this mode of operation, all of the instrument's sound is generated within the instrument itself, by the built in loudspeakers. This is closest to how an acoustic instrument such as the acoustic guitar works.

Internal and external sound production: this mode offers two sound sources. The first is the instrument itself, through the embedded loudspeakers. The second source is an external (and possibly amplified) loudspeaker. This mode of operation is based on instruments such as the electric guitar or electric violin.

Modified internal sound production: when using both internal and external sound production it is possible to modify the signal used for internal sound production, creating a difference between the sound created internally by the instrument itself and that produced by the external system.

When using different signals for each sound generating mechanism, we can perform a number of interesting effects, including:

- Compensation for the frequency response of the loudspeakers and/or human skin (as in [3]).

- Simulation of the frequency response of a different instrument.
- Production of only those frequencies for which the skin is sensitive.

Each of these effects can be performed for the internal sound production and vibrotactile feedback portion of the instrument, while still producing the unmodified sound from the sound synthesis system through the external sound production mechanism.

When producing vibrotactile feedback it is interesting to note that neither the actuators used to produce vibrotactile feedback nor the human skin offer a flat response to vibrations across the frequency range. By having separate control over the frequency content of the signal sent to the vibrotactile feedback system we can compensate for these responses. For instance, if the instrument is to generate low frequency sounds it is possible that the loudspeakers used may have a reduced response at these frequencies. By modifying the signal sent to the loudspeakers we could increase the output amplitude for these low frequencies.

Modification of the vibrotactile feedback signal can also be used to modify the vibration response in such a way as to make it more like the response of a different instrument. It is possible to increase or reduce the response at certain frequencies or within certain frequency bands. This could, for instance, be used to produce low frequency vibrations for an instrument with a poor low frequency response. It could also be used, together with measurements of the vibration response of an existing musical instruments, to simulate the resonances of the body of other instruments in the Viblotar.

Finally, by modifying the feedback signal, we can restrict the sound produced by the internal sound production mechanism (and thus the vibrations created) to only those frequencies to which the human skin is sensitive. This results in the internal sound production being used mostly for vibration production, while the actual sound production occurs outside of the instrument itself. In fact, it would even be possible to restrict the internal sound production to frequencies which are too low to be audible, thus using it solely for vibration generation.

It is also possible (and perhaps even advisable) to combine a number of these effects together. For instance, when attempting to simulate the resonances of another instrument it may well be necessary to apply compensation for the actuator so that the target response is produced by the system.

3. MEASURING INSTRUMENT VIBRATIONS

For some of the effects just discussed, and indeed to enable a mechanical evaluation of the vibrotactile feedback system used in the Viblotar, it is necessary to be able to measure the vibrations of a given instrument, whether acoustic or digital. This section describes a method of measuring instrument vibrations and provides examples and comparisons of the vibration of an acoustic guitar and the Viblotar. The measurement method described in this section is based on that used by [2], who measured the vibration response of a number of stringed instruments at different points on the instrument body.

The aim of the measurements made here are to compare the vibrations of an acoustic instrument (an acoustic steel stringed guitar) with a new digital musical instruments (the Viblotar). In particular, we are interested in showing certain common traits between these two different instruments. Questions of particular interest are:

1. Do these instruments produce vibrations above the threshold of human detection?
2. Are there similarities in the spectral content of these vibrations?
3. Are the spectra of the vibrations related to the note being played?

3.1 Methods and Procedure

All vibration measurements were made with the instrument in normal playing position. A PCB Piezotronics ICP accelerometer, model 352C22 was used for all vibration measurements. The output signal from the accelerometer was connected to a PCB Piezotronics ICP Signal Conditioner, model 480E09. Analog to digital conversion of the amplified voltage was performed using a National Instruments PCI-6036E with a 16-bit resolution and a sampling rate of 100 kHz. Finally, control and datalogging was performed using National Instruments LabView 7.1 software. Analysis of the recorded signals was performed with Matlab.

Figure 2: The Viblotar in the playing position.

For each instrument, the accelerometer was attached at the measurement position using adhesive wax. Each instrument was held in the playing position. All measurements were performed using a single pitch, corresponding to the open low E string of the guitar. This gives a frequency of 82 Hz. Multiple measurements were made for each instrument. These measurements were averaged during the analysis stage to reduce the effect of any artefacts from single notes.

For the guitar, the procedure was as follows: the accelerometer was attached to the instrument on the top plate, near the bridge. The instrument was held in the playing position, with the neck resting in the left hand, but no fingers pressed to the fingerboard. The low E string was plucked using a pick at the specified dynamic level and allowed to resonate until no detectable vibrations were present. This was repeated 10 times.

For the Viblotar, the procedure was similar. The instrument was held in the playing position, with the body of the instrument resting on the performer's legs, as shown in Figure 2. The left hand was allowed to rest on the left side of the instrument, near the Force-sensing resistors (FSRs). The right hand was also allowed to rest on the instrument, directly below the linear position sensor. For the purpose of this experiment, the Viblotar mapping was modified so that a touch at any point on the sensor produced the desired note. The linear position sensor is touched using one of the

fingers of the right hand. The note is allowed to resonate until no detectable vibrations are present. To ensure no accidental damping or modulation of the note occurs, these functions of the mapping system were also disabled for the duration of the test. As with the guitar, this procedure was repeated 10 times.

3.2 Results

Figure 3 shows the average vibration spectrum measured for the acoustic steel string guitar. Notice the peaks fundamental and each of its harmonics. The spectrum shows especially large peaks at the 2^{nd} and 4^{th} harmonics. Note also how the vibrations in the lower frequencies are above the threshold of human vibrotactile detection.

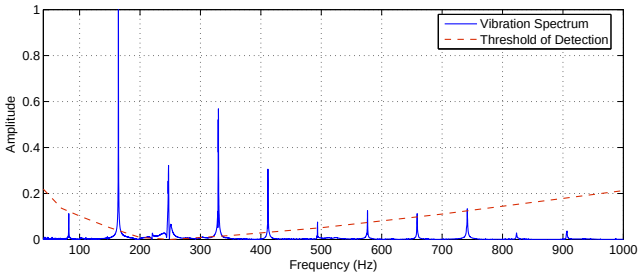


Figure 3: Average vibration spectrum of an acoustic steel string guitar playing open low E (82 Hz), as measured near the bridge.

The average vibration spectrum for the Viblotar is shown in Figure 4. As with the guitar, it shows peaks at the harmonics of the note played. Unlike the guitar, there are also peaks in the spectrum at non-harmonic frequencies. These peaks are due to the flute portion of the hybrid guitar/flute model used in the blotar synthesis. Similar to the guitar, the lower frequencies are above the threshold of detection. Unlike the guitar, a number of higher frequencies are also well above the threshold of detection.

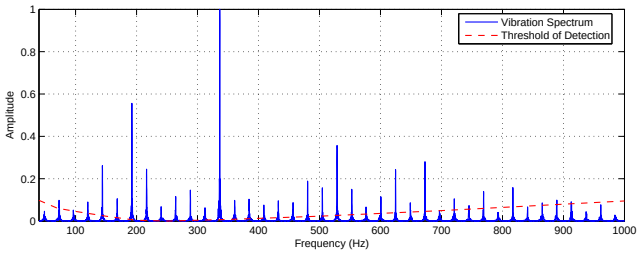


Figure 4: Average vibration spectrum of the Viblotar playing a frequency of 82 Hz, as measured on the top

Examining both spectra, it can be seen that both instruments produce vibrations above the threshold of detection. There are also a number of similarities in the spectra, each producing detectable vibrations at a number of frequencies which are harmonics of the note being played.

Having examined the vibrations produced by these instruments, we can see that both produce vibrations which would be felt by the performer. Also, the vibrations produced by the Viblotar are similar to those produced by an acoustic instrument. This then raises the question of whether these vibrations affect the "feel" of the Viblotar for the performer. The experiment described in the next section attempts to deal with this question.

4. EXPERIMENT: PERFORMER EVALUATION

This section describes an experiment to evaluate the effects of the embedded vibrotactile feedback system on the "feel" of the Viblotar. While the concept of the "feel" of an instrument is one which is often mentioned by performers it is difficult to objectively evaluate. Therefore, for this experiment a measure of the "feel" of the instrument is determined based on a number of different characteristics, which participants are asked to rate:

Ease of use: how easy the instrument is to perform with.

Controllability: how much the performer was in control of the instrument.

Engagement: how much of the performer's attention was put into playing the instrument.

Entertainment: how entertaining the instrument is.

Potential for further performance: how much potential the instrument offers for further performance.

4.1 Participants

The participants were 5 graduate students from McGill University. All participants were experienced musical performers, having completed at least an undergraduate degree in music performance. Two of the participants had previous experience playing digital musical instruments, while the others did not. None of the participants were familiar with the Viblotar.

4.2 Design and Materials

The aim of this experiment was to examine how the choice of sensors and feedback affected the "feel" of the instrument. To evaluate this we asked performers to play the Viblotar in two different configurations:

1. With external sound production and no vibrotactile feedback.

2. With internal sound and vibrotactile feedback production.

In the external sound production configuration, the synthesized sound is output using a pair of loudspeakers which are placed in front of the performer at a distance of 1 meter. This removes all vibrotactile feedback from the instrument and dissociates the sound from the instrument itself. The result of this is a configuration like existing digital musical instruments.

With the internal sound production, the sound is produced using the two loudspeakers which are in the body of the instrument itself. This results in vibrotactile feedback to the performer and in the sound coming from the instrument in a way most like an acoustic instrument. For both configurations the sound volume was maintained at the same level (90dB peak, A-weighted), measured using a Radio Shack 33-2055 digital SPL meter.

These configurations allow for an examination of the effects of vibrotactile feedback and embedded sound production on performer ratings of the instrument.

Overall, the hypothesis for this experiment is that Vibrotactile feedback should improve the "feel" of the instrument. This means that some performer ratings should be higher for the internal sound production configuration.

4.3 Procedure

Subjects arrived at the lab and were given an Information/Consent form to read over and sign. Subjects were then introduced to the Viblotar and its playing interface. The sensors used on the Viblotar were explained, along with the parameters that they control. They were then given a demonstration of playing the instrument.

Subjects were informed that they would be playing the instrument in two different configurations, although they were not told what the difference between each configuration was. They were told that for each configuration they would be allowed to play the Viblotar for 20 minutes and then asked to rate the instrument on several criteria. They were shown the list of criteria and each item was explained to them. The order of presentation of the configurations was randomized. All ratings were performed on a 5-point Likert scale.

Participants then spent 20 minutes performing with the instrument in the first configuration. Once the time was up, they rated that configuration on each of the criteria being examined. This process was then repeated for the second configuration.

Finally, participants were debriefed verbally after the experiment and asked for any comments they had on the instrument or either configuration. The differences between each configuration was also explained at this point.

4.4 Data Analysis

Results were analyzed in Matlab. As the data was found not to follow a normal distribution the analysis was performed using the Wilcoxon signed rank test.

4.5 Results

There was a marginally significant improvement in engagement for the configuration with vibrotactile feedback [p = .07] (Figure 5). This was the only significant difference found in this experiment. However, there were also two non-significant differences found between configurations.

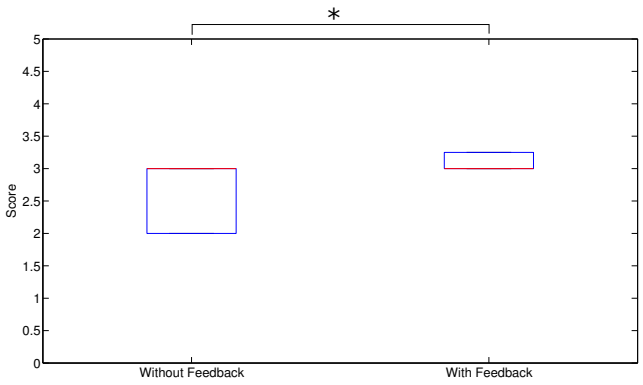


Figure 5: Participant ratings of engagement with the Viblotar, with and without vibrotactile feedback. A * indicates a significant difference. Red lines indicate median values, while blue lines indicate lower and upper quartile values. Whiskers extend to 1.5 times the interquartile range.

Firstly, there was a slight improvement in entertainment ratings for the vibrotactile feedback configuration [$M_{without}$ = 3.0, M_{with} = 3.4] (see Figure 6). In contrast to this, there was a slight deterioration in ratings of the controllability of the instrument for the vibrotactile feedback configuration [$M_{without}$ = 3.8, M_{with} = 3.4] (see Figure 7).

There were no significant differences in user ratings of the configurations for ease of use or potential for future performance.

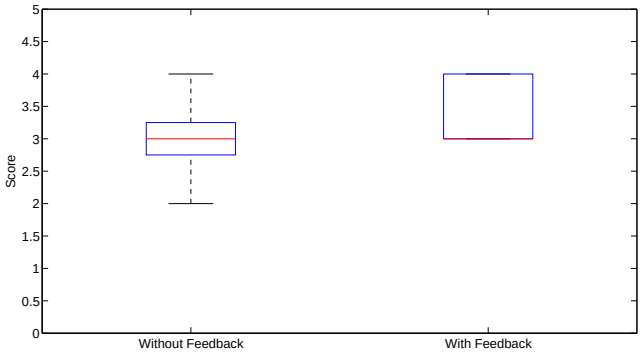


Figure 6: Participant entertainment ratings of the Viblotar, with and without vibrotactile feedback.

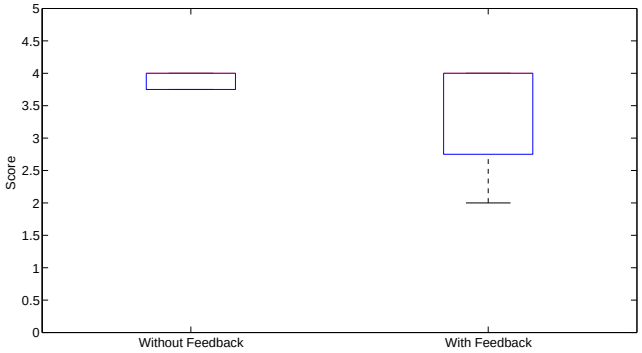


Figure 7: Participant ratings of the controllability of the Viblotar, with and without vibrotactile feedback.

Interestingly, ease of use ratings were high [M_{ease} = 4.4] in both configurations. In fact, the ratings were identical for both configurations. This indicates that while the instrument is easy to use, the ease of use is not in any way affected by the addition of vibrotactile feedback. This may be an artefact of the design of the instrument itself, as previous work has shown that vibrotactile feedback can make an instrument easier to use [10].

4.6 Discussion

A number of interesting points arise from the results of this experiment. Firstly, the ease of use ratings for both configurations were high. A mean ease of use of 4.4 out of 5 was received by each configuration. This indicates that the sensors chosen provide an easy to use interface. The fact that each participant gave the same ease of use rating for both configurations would also seem to confirm that this result is due to the combination of sensors, gestures and tasks, as it was unaffected by the presence or absence of vibrotactile feedback.

Looking at the effects of vibrotactile feedback, we find a number of criteria which change when this feedback is present. Firstly, there was a marginally significant improvement in engagement when feedback was present [t(4) = 2.45, p = .07]. Participants found themselves more engaged with the instrument when vibrotactile feedback was present. They were more involved in the performance of the instrument, spending more of their attention on the instrument.

Interestingly, participant rating of controllability dropped with the addition of vibrotactile feedback [$M_{without}$ = 3.8, M_{with} = 3.4]. Participants felt less in control of the in-

strument when the feedback was present. One participant commented on noticing changes in the sound for the internal sound production configuration that had not been noticed for the other configuration. This could indicate that the vibrotactile feedback channel was providing extra information to the performers that was not present in the other configuration, so that they noticed changes which they would otherwise have missed. Such extra information could be extremely useful for developing expert performance technique. It is also possible to consider that a reduction in controllability might result in an increase in the challenge involved in performing the instrument. This could have an effect on the overall performance potential of the instrument in the longer term.

Finally, there was a small increase in entertainment ratings for the configuration with internal sound and feedback generation [$M_{without} = 3.0$, $M_{with} = 3.4$]. Together with the significant increase in engagement this would seem to indicate that the playability, or indeed the "feel" of the instrument is improved when vibrotactile feedback is present.

5. CONCLUSIONS

The work described in this paper examined the use of embedded vibrotactile feedback in a digital musical instrument and it's effect on the "feel" of the instrument from the performer's perspective. By integrating loudspeakers and amplifiers in to the body of the Viblotar, we produced an instrument that mimics the vibrotactile feedback found in acoustic instruments. That is, the sound production also produces the vibrotactile feedback.

The addition of internal sound generation to the Viblotar produced a number of effects. It localized the sound to the instrument itself and it added vibrotactile feedback to the instrument. Looking at the results of the experiment in Section 4, we can see that this resulted in a marginally significant increase in engagement, along with a small (although not significant) increase in entertainment. This would seem to indicate that there is an improvement in the "feel" of the instrument for the performer when vibrotactile feedback is present.

Interestingly, the additional vibrotactile feedback also resulted in a slight (and again not significant) decrease in performer controllability ratings. In post-experiment debriefing, one of the participants explained that they thought the sound synthesis had changed between configurations. On further examination it was discovered that the participant had noticed changes in the sound under the vibrotactile feedback configuration which had not been noticed under the other configuration. More information was being presented to the performer by the extra feedback channel. It seems that this extra information was causing the performer to feel less in control of the instrument than in the other configuration.

However, this raises some interesting issues. Wessel and Wright state that a musical instrument should offer a "low entry fee" but with "no ceiling on virtuosity" [12]. Instruments which are too easy to use may seem more like toys and less like instruments. Hunt found that users enjoy performing with instruments which offer more of a challenge [6]. For the Viblotar, the addition of vibrotactile feedback resulted in reduced controllability ratings. This might indicate that the instrument becomes more challenging with the feedback present, as it provides more information about the state of the instrument to the performer.

However, a number of issues still remain to be addressed. A longer term evaluation, perhaps with more participants, could lead to much insight into the playability of the Viblotar. Keele states that vibrotactile feedback is used more by expert performers than beginners [7]. As the participants in the experiment in this study were all novice Viblotar players, it is possible that they were not making use of the vibrotactile feedback in the same way as an expert performer would. A longer term experiment examining the changes in user ratings over a longer period of time would allow the participants to increase their skill with the instrument. Such an experiment might also lend insight into the effects of the vibrotactile feedback on the "feel" of the instrument, through changes in participant ratings over time.

6. REFERENCES

[1] N. Armstrong. *An Enactive Approach to Digital Musical Instrument Design*. PhD thesis, Princeton University, Nov. 2006.

[2] A. Askenfelt and E. Jansson. On vibration sensation and finger touch in stringed instrument playing. *Music Perception*, 9(3):311–350, 1992.

[3] D. Birnbaum and M. M. Wanderley. A systematic approach to musical vibrotactile feedback. In *Proceedings of the 2007 International Computer Music Conference (ICMC07)*, volume II, pages 397–404, Copenhagen, Denmark, 2007.

[4] C. Chafe. Tactile audio feedback. In *Proceedings of the 1993 International Computer Music Conference (ICMC93)*, pages 76–79, Tokyo, Japan, 1993.

[5] P. R. Cook. Remutualizing the musical instrument: Co-design of synthesis algorithms and controllers. *Journal of New Music Research*, 33(3):315–320, Sept. 2004.

[6] A. Hunt. *Radical User Interfaces for real-time musical control*. PhD thesis, University of York, 2000.

[7] S. W. Keele. *Attention and Human Performance*. Goodyear Publishing Company, 1973.

[8] M. Marshall. *Physical Interface Design for Digital Musical Instruments*. PhD thesis, McGill University, 2009.

[9] M. T. Marshall and M. M. Wanderley. Vibrotactile feedback in digital musical instruments. In *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06)*, pages 226–229, Paris, France, 2006.

[10] S. O'Modhrain. *Playing by Feel: Incorporating Haptic Feedback into Computer-Based Musical Instruments*. PhD thesis, Stanford University, 2000.

[11] M. Puckette and Z. Settel. Nonobvious roles for electronics in performance enhancement. In *Proceedings of the 1993 International Computer Music Conference (ICMC93)*, pages 134–137, Tokyo, Japan, 1993. International Computer Music Association.

[12] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

HIDUINO: A firmware for building driverless USB-MIDI devices using the Arduino microcontroller

Dimitri Diakopoulos[1]
California Institute of the Arts[1]
24700 McBean Parkway
Valencia, California 91355
ddiakopoulos@alum.calarts.edu

Ajay Kapur[1, 2]
New Zealand School of Music[2]
P.O. Box 2332
Wellington, New Zealand
akapur@calarts.edu

ABSTRACT
This paper presents a series of open-source firmwares for the latest iteration of the popular Arduino microcontroller platform. A portmanteau of Human Interface Device and Arduino, the HIDUINO project tackles a major problem in designing NIMEs: easily and reliably communicating with a host computer using standard MIDI over USB. HIDUINO was developed in conjunction with a class at the California Institute of the Arts intended to teach introductory-level human-computer and human-robot interaction within the context of musical controllers. We describe our frustration with existing microcontroller platforms and our experiences using the new firmware to facilitate the development and prototyping of new music controllers.

Keywords
Arduino, USB, HID, MIDI, HCI, controllers, microcontrollers

1. INTRODUCTION
A core goal of the Music Technology program at the California Institute of the Arts is to teach students how to connect the physical and virtual world. *Interface Design for Music and Media Applications* is a yearlong class that introduces students to artistic interactivity through microcontrollers and sensors/actuators. Modeled after the template presented at CCRMA by Bill Verplank *et al* in 2001, "A Course on Controllers [11]," and Gideon D'Arcangelo's course at ITP [3], the class required a modern microcontroller platform that was neither too high-level nor too complex.

The Arduino turned out to be the ideal solution for the class, although there was a major usability issue which we felt restricted its potential as the core of a musical controller, described later in section 2.2. Combined with a redesign of the Arduino platform in 2010 and an open-source USB communication library for Atmel AVR microcontrollers (on which the Arduino is based), we were able to develop a firmware permitting driverless USB-MIDI communication between Arduino and host computer, a solution that resolved our greatest usability concern with the Arduino.

In section 2, we present our reasons for switching to the Arduino, some other platforms aimed at artistic interactivity, and frustrations with both. Section 3 details the nuts and bolts of the HIDUINO firmware. Section 4 explains the process of prototyping a controller using HIDUINO. We conclude in Section 5 with several ideas about future improvements.

2. BACKGROUND
The third iteration of the *Interface Design* class in 2009 was the first to switch to the Arduino[1]. It was during this time we became acutely aware of the power of the platform, but also its shortcomings, namely that serial data needed to be parsed and converted to a more useful format. While the complete history of using microcontrollers in the context of NIME is outside the scope of this paper, we provide a brief overview of some related projects that attempt to solve this protocol problem.

2.1 Why the Arduino
The class follows the basic ideas Perry Cook outlines in "Designing Principles for Computer Music Controllers [1, 2]," as well several requirements described by Nicola Orio *et al* in [5], including *learnability, exportability,* and *feature compatibility.* Our selection of a suitable microcontroller for the class held these concepts in mind, applying them not only to the properties *of* a musical controller but also to the elements core to their design. Based on the research presented by Scott Wilson *et al* in [12], the results of using Atmel's AVR microcontroller appeared to meet many of these design criteria.

Since the publication of that paper, many AVR-based microcontroller platforms have been released, including the increasingly popular Arduino. Our choice to move to this platform was motivated by the large community of support and open-source nature of many Arduino-based projects. In Alicia Gibb describes the Arduinos' growing reputation as an extensible platform for interactive media and goes on to say, "The design of the Arduino microcontroller caters to a non-technical audience by focusing on usability to achieve its intended goal as a platform for designers and artists [4]," supporting one of our core criteria of *learnability.* As the Arduino language is simply an abstracted form of C, we found that our *exportability* and *feature-compatibility* requirements were sufficiently satisfied by the ability to write and use low-level C libraries for more advanced projects.

2.2 On Protocol Confusion & Usability
Hans-Christoph Steiner's paper, "Firmata: Towards making microcontrollers act like extensions of the computer [9]," reveals a general problem in existing microcontroller platforms: protocols for communication. Arduino, and other similar platforms like Wiring[2] and Gainer[3], implement Virtual COM ports via USB for basic serial I/O between microcontroller and host. Since no major music application outside of Max/MSP[4] supports reading serial directly, there is a significant disconnect between controller and application.

[1] http://www.arduino.cc/
[2] http://wiring.org.co/
[3] http://gainer.cc/
[4] http://cycling74.com/

For us, this single limitation created a large usability gap in the platform which turned into the primary motivating factor behind HIDUINO.

It was necessary in the 2009 CalArts *Interface Design* class that each new musical controller required a new Arduino sketch and accompanying 'decoder' software to interpret the raw serial data and convert to music-friendly MIDI or OSC. MIDI was particularly problematic since it required the use of virtual MIDI loopback drivers (or proprietary loopback software, in the case of Windows). Considerable time was added to the development of controllers on account of the need for this middleware software. Additionally, it created a single point of failure and added additional latency between performer and application. Figure 1 illustrates an overview of a controller using this complicated method.

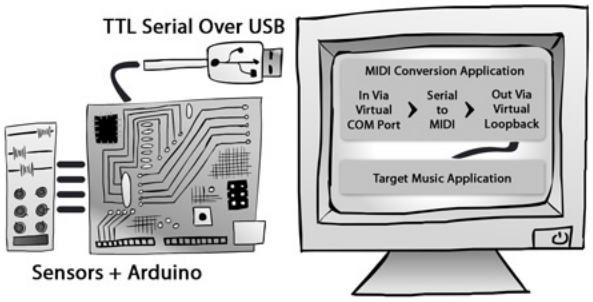


Figure 1. Controller Development Pre-HIDUINO

Our initial attempts at simplifying the process led us to look to preexisting solutions in the Arduino community. Sidestepping the need for serial and separate software, we sound several potential solutions in the form of external hardware add-ons known to the Arduino community as *shields.* We evaluated two shields, one for MIDI[5], and another for Ethernet[6].

In the case of the MIDI shield, we noted that the host computer still needed a separate MIDI interface for the older-style 5-pin DIN connection and was additionally limited by all the typical constraints of MIDI. The Ethernet shield we tested was used in conjunction with a simple OSC implementation. This combination added additional cost and extra cabling: USB is required for power, Ethernet for data. Moreover, many students wishing to use their controller with commercial software still needed an OSC to MIDI conversion app.

2.3 USB HID & the CUI

The vast majority of commercial MIDI controllers on the market implement a protocol known as USB-HID[7]. This protocol is often viewed negatively by developers, citing its implementation complexity and bloat [9]. On account of these difficulties, few have been able to implement the protocol in a working form suitable for musical controller development. However, one of the more recent and successful projects using USB-HID is the Create USB Controller (CUI) developed by Dan Overholt at UC Santa Barbara [6].

Prior to the 2009 *Interface Design* class, the CUI was the main controller platform on account of its native USB-HID support. Built on top of a Microchip PIC[8] (a competitor to the Atmel AVR), the CUI comes bundled with a generic MIDI-HID firmware which sends out 12-channels of pitch-bend data, one for each ADC pin present on the board.

In previous years, students in the class noted a few frustrations with the CUI, claiming the propriety development chain and necessity of low-level C required by Microchip complicated the process of making changes to the firmware. We also investigated the use of the micro-OSC firmware on the CUI, uOSC [7, 8]. The uOSC firmware also required separate serial to OSC software on the host. In short, it did not meet our expectation of usability.

Figure 2 illustrates the benefits of the USB-HID protocol by simplifying the number of steps present in Figure 1 and demonstrates our ideal model for a musical controller development platform.

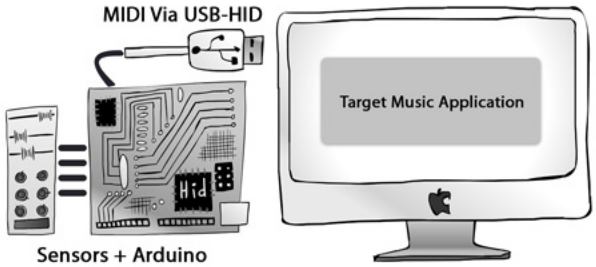


Figure 2. Simplification of protocols and software by using HID.

2.4 A New Firmware

Underling several ideas presented by Owen Vallis *et al* in his 2010 NIME presentation, "A Shift toward Iterative and Open-Source Design for Musical Interfaces [10]," the concept of HIDUINO aimed to directly attack our usability issues by removing the protocol confusion. Driven by a desire to see the Arduino act as a true USB-HID device, the primary goal of HIDUINO was to dismiss the need for custom software and remove the Arduino's dependence on the serial protocol. A secondary goal was to develop a feature-complete framework to meet the needs of both prototype and performance-ready controllers.

3. IMPLEMENTATION

Implementation of HIDUINO was aided by two recent events within the Arduino community: a redesign of the Arduino microcontroller and the support of an existing USB-HID library by the Arduino team.

3.1 2010 Arduino Redesign

The 2010 revision to the Arduino platform introduces the UNO and the Mega2560, using the Atmel ATmega328 and the ATmega2560 chips respectively. Earlier revisions were equipped with an FTDI chip permitting users to interface with the Arduino via USB. The FTDI chip presented a few challenges to familiar users, namely that it required propriety drivers on all platforms and could *only* act as a virtual serial port. The 2010 redesign omitted this chip in favor of the ATmega 8U2, the tiniest chip in Atmel's lineup that included native support for USB. The new Arduino includes a pre-loaded firmware which emulates the functionality of the older FTDI chip, but more importantly exposes the pins necessary to re-flash the 8U2 with the users' own custom firmware. This change opened up the possibility of writing a firmware that could conform to the USB-HID protocol specification and still communicate with the primary microcontroller on the Arduino. Without this redesign, the HIDUNO project would have required the production of a new shield that incorporated the 8U2 or similar ATmega USB chip.

[5] http://www.sparkfun.com/products/9595
[6] http://www.arduino.cc/en/Main/ArduinoEthernetShield
[7] http://www.usb.org/developers/hidpage/
[8] http://www.microchip.com/

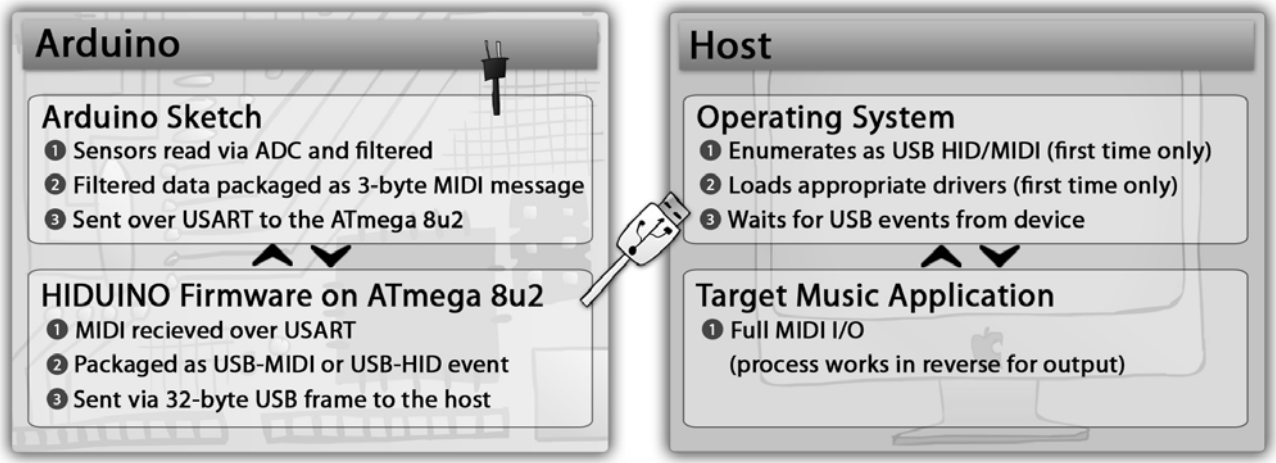


Figure 3. High-level architecture of a controller using HIDUINO

3.2 LUFA Library

In 2008 Dean Camera started the MyUSB library, an open-source Human Interface Device (HID) library for the USB-compatible line of AVR microcontrollers. Later renamed LUFA (Lightweight USB Framework for AVRs)[9], the project set out to create an elegantly-written library to demystify the USB-HID protocol. In contrast with a number of libraries written for the same purpose (detailed in the LUFA documentation)[10], the API is straightforward and not restricted to any specific AVR USB microcontroller. In addition, the library comes preloaded with descriptors for generic HID devices, including MIDI. Descriptors, as a core part of the USB protocol, instruct which drivers a host computer should use to interface with the device. Most operating systems provide built-in drivers for these USB *class-compliant* devices.

3.3 The Firmware(s)

Initial firmware programming took place late in 2010 shortly after the new Arduino designs shipped. The LUFA library already provides appropriate descriptors for USB-MIDI, thus the majority of HIDUINO code is targeted at structuring the communication between the main ATmega chip and the new 8U2. Figure 3 illustrates a full-system overview of a controller using the HIDUINO, showing the primary functions of the firmware.

The main Arduino ATmega and 8U2 chip communicate over shared transmit/receive USART[11] pins. Although the structure of the serial sent from the main chip does not matter on account of its later re-packaging as 32-byte USB-MIDI event, we decided implement a standard 3-byte MIDI protocol to standardize and simplify sending data between the chips. Once the HIDUINO firmware receives a complete 3-byte message, it is checked for validity, placed into a USB-MIDI event container, and finally pushed over USB. After developing MIDI-out, all communication functions were re-written in reverse for MIDI-in.

While we emphasize development of USB-MIDI in this publication, other HIDUINO firmwares are currently being released that give an Arduino the power to act like other common HID devices, including mice, keyboards, joysticks, game controllers, and even audio devices. Several music programming languages and frameworks currently support reading HID, though these are aimed primarily at repurposing commercial USB devices as musical controllers [13].

4. PROTOTYPING

The process of building a controller implementing HIDUINO is designed to be as straightforward as possible. In the context of the *Interface Design* class, the process of prototyping a new controller can be broken down into four steps:

1. Design
2. Signal Conditioning
3. Transitioning to HIDUINO
4. Testing

4.1 Design

The design phase exists to consider aspects of overall form, function, and effectiveness as musical controller. Students are encouraged to test and experiment with various sensors including potentiometers, soft potentiometers, force sensing resistors, buttons, accelerometers and gyros, photocells, resistive touch surfaces, proximity sensors, hall-effect sensors, and flex sensors.

4.2 Signal Conditioning

All sensors connected to the Arduino require some level of conditioning and scaling before being sent to the host. For example, accelerometers are often low passed and soft-pots are read using sample-and-hold logic. Data during this step is sent to the host using serial so it can be displayed in the serial monitor in the Arduino IDE. As a final step, sensor data is clamped to MIDI-friendly 0-127 resolution.

4.3 Transitioning to HIDUINO

After a user is content with the sensor data, the Arduino sketch is ready to implement MIDI. Using a simple Arduino library for reading and writing MIDI over serial, the data can be packaged as a note on, continuous control, or pitch bend message.

4.3.1 Flashing the firmware

The flashing process can be accomplished one of two ways depending on whether a user has access to an in system programmer (ISP). Both ATmega chips on the redesigned Arduino have exposed in circuit serial programming (ICSP) headers. In our own testing and within the class, an Atmel AVR-ISP MKII was utilized to flash firmwares directly onto the 8U2. A second option is through the use of a bootloader.

[9] http://code.google.com/p/lufa-lib/
[10] http://www.fourwalledcubicle.com/files/LUFA/Doc/101122
[11] http://arduino.cc/en/Reference/serial

The DFU bootloader[12] written by Atmel can piggyback on top of any firmware granted there is enough free flash memory. When certain pins on the exposed 8U2 headers are tripped, the chip will enter bootloader mode and then can be programmed via USB. All HIDUINO firmwares are currently built with the DFU bootloader so both methods are available. Section 5 provides a link to the HIDUINO project page which presents this entire process in greater depth (including tools and software used) in tutorial format.

In the first iteration of HIDUINO, host communication with the primary ATmega328 chip needed the virtual-serial port firmware and thus complicated the process of updating Arduino sketches as most users need to continuously switch between virtual-serial and HIDUINO. A recent build of the software combines the virtual-serial and HIDUINO firmwares into a single package so a user is able to select which firmware is loaded based on header pin configuration. This build requires the use of an ISP programmer to initially load the firmware as the combined version could not be combined with the DFU bootloader in the 8kB of space available on the 8U2.

4.4 Testing

The testing phase ensures that each sensor is correctly scaled and addressed by the right MIDI identifier.

5. SUMMARY AND FUTURE WORK

The HIDUINO project represents a significant step forward for students, musicians, and artists who desire their own custom controller. With HIDUINO, the Arduino now has the potential to become a powerful base for driverless, cross platform MIDI controllers and other HID devices. With these firmwares, DIY controllers can compete with the plug-and-play usability of commercial offerings while maintaining our core values of learnability, modularity, and flexibility for teaching and prototyping.

With respect to future work, considerable ongoing effort is being applied toward extending the quality and number of HID firmwares, including device types for joysticks, game controllers, mice, and keyboards. One of the largest student complaints about the MIDI firmware is that it is still restricted to 127[13] steps of resolution. We are investigating the possibility of implementing a part of the USB specification called CDC-ECM[14] – Ethernet Control Model. Although not currently a part of the LUFA library, CDC-ECM would allow the possibility of native Ethernet-over-USB functionality permitting the use of high-resolution protocols like OSC.

The HIDUINO project page is located online at http://mtiid.calarts.edu/research/hiduino and includes a tutorial-style guide to alter and compile the firmwares from scratch. The current SVN code repository is located on GoogleCode at http://code.google.com/p/hiduino/.

6. ACKNOWLEDGMENTS

This project would have been very difficult without the Lightweight USB Framework for AVRs (LUFA) written by Dean Camera. The authors would also like to thank Martijn Zwartjes, Jim Murphy, and the entire Arduino team and community. Special thanks to Tahnee Gehm for illustrating Figures 1, 2 and 3.

7. REFERENCES

[1] Cook, P.R. "Principles for Designing Computer Music Controllers," in *ACM SIGCHI New Interfaces for Musical Expression (NIME) Workshop* Seattle, WA, 2001.

[2] Cook, P.R. "Re-Designing Principles for Computer Music Controllers: a Case Study of SqueezeVox Maggie," in *New Interfaces for Musical Expression (NIME)* Pittsburgh, PA, 2009.

[3] D'Arcangelo, G. "Creating a Context for Musical Innovation: A NIME Curriculum," in *New Interfaces for Musical Expression (NIME)* Dublin, Ireland, 2003.

[4] Gibb, A.M. *New Media Art, Design, and the Arduino Microcontroller: A Malleable Tool.* Master's Thesis, Pratt Institute, New York, NY, 2010.

[5] Orio, N., Schnell, N., and Wanderley, M.M. "Input Devices for Musical Expression: Borrowing Tools from HCI," in *Computer Music Journal*, 26(3), MIT Press, 2002.

[6] Overholt, D. "Musical Interaction Design with the CREATE USB Interface: Teaching HCI with CUIs instead of GUIs," in *International Computer Music Conference (ICMC)* New Orleans, LA, 2006.

[7] Schmeder, A. and Freed, A. "A Low-level Embedded Service Architecture for Rapid DIY Design of Real-time Musical Instruments," in *New Interfaces for Musical Expression (NIME)* Pittsburgh, PA, 2009.

[8] Schmeder, A. and Freed, A. "uOSC: The Open Sound Control Reference Platform for Embedded Devices," in *New Interfaces for Musical Expression* (NIME) Genova, Italy, 2008.

[9] Steiner, H.C. "Firmata: Towards making microcontrollers act like extensions of the computer," in *New Interfaces for Musical Expression (NIME)* Pittsburgh, PA, 2009.

[10] Vallis, O., Hochenbaum, J., and Kapur, A. "A Shift Towards Iterative and Open-Source Design for Musical Interfaces", in *New Interfaces for Musical Expression (NIME)* Sydney, Austalia, 2010.

[11] Verplank, B., Sapp, C., and Mathews, M. "A Course on Controllers," in *ACM SIGCHI New Interfaces for Musical Expression (NIME) Workshop* Seattle, WA, 2001.

[12] Wilson, S., et al. "Microcontrollers in Music HCI Instruction", in *New Interfaces for Musical Expression (NIME)* Montreal, Canada, 2003.

[12] DFU bootloader datasheet: http://www.atmel.com/dyn/resources/prod_documents/doc7618.pdf

[13] Assuming MIDI pitch bend is not used

[14] CDC-ECM Specification: http://www.usb.org/developers/devclass_docs/CDC_EEM10.pdf

Latency improvement in sensor wireless transmission using IEEE 802.15.4

Emmanuel Fléty, Côme Maestracci
IRCAM - Real Time Musical Interactions
STMS IRCAM - CNRS - UPMC
1 Place Igor Stravinsky
75004 - Paris - France
emmanuel.flety@ircam.fr

ABSTRACT
We present a strategy for the improvement of wireless sensor data transmission latency, implemented in two current projects involving gesture/control sound interaction. Our platform was designed to be capable of accepting accessories using a digital bus. The receiver features a IEEE 802.15.4 microcontroller associated to a TCP/IP stack integrated circuit that transmits the received wireless data to a host computer using the Open Sound Control protocol. This paper details how we improved the latency and sample rate of the said technology while keeping the device small and scalable.

Keywords
Embedded sensors, gesture recognition, wireless, sound and music computing, interaction, 802.15.4, Zigbee.

1. INTRODUCTION
The use of wireless sensor data transmission has been used for several decades in computer music either as a sensing technique by itself [8][11] or as digital transmission media [3][1].

Wireless solutions are now even widely used in interactive gaming with devices like the Nintendo Wii remote or Sony Playstation Move controller.

Those ready-made devices can be extensively hacked and modified to serve the purpose of musical interaction, however they suffer from a clear lack of performance in the sensor sampling rate and/or the latency jitter, two aspects leading to a major impact on the interaction quality [10].

Our goal was to improve our current wireless system that uses off-the-shelf Xbee zigbee modules (presented in previous NIME[2]). XBee modules are popular and feature suitable performance, allowing multi-node sensor digitizers as found in the Sense/Stage Project[7]. However, their main drawback is a locked firmware which does not allow a seamless configuration of each node. We therefore specifically redesigned our hardware into a more generic platform for enhanced performances. This research was performed in the context of *Interlude* and *Urban Musical Game* projects where gestural interface were built for either collaborative music playing or music pedagogy [9].

2. KNOWN ISSUES
Designing a wireless system presents several known issues and challenges such as CSMA/CA when the media is shared amongst several nodes (star or mesh networks) and the implementation of the numerous layers of the OSI model.

Turnkey radio modules are affordable and easy to implement with custom electronics including usually a small microcontroller unit (MCU) acquiring the sensors' data. However forwarding the data to the radio module has an important impact (slow UART) on the system latency which could be significantly improved.

Widely used for experiments, single radio channel transmitter/receiver pairs (wireless UARTs) were legion a decade ago with the known drawback of permanently occupying the radio channel, making the simultaneous use of multiple nodes on stage impossible without using several radio channels which were highly dependant on local FCC regulations.

In order to allow multiple performers on stage, we therefore oriented our past designs toward actual networking technology such as 802.11 [4]. This was made easier thanks to the emergence of the worldwide 2.4 GHz ISM band for general purpose broadcasting and dedicated wireless networking which was unfortunately bulky and featured a limited runtime.

Bluetooth appeared as a promising, power friendly solution. It is still widely used for hacking but it has not received a positive echo for live performance use despite a fairly high data rate as pointed out by Torrensen [12]. Bluetooth suffers from a significant jitter in the transmission latency[1] and its packetizing effect results in the loss of the original timing of the sampled data. This usually worsens when a Bluetooth module is directly paired with a host computer since the software Bluetooth stack is not designed for real-time applications and features unmanageable time-outs, making the wireless link impossible to reset safely during the performance.

To overcome these issues, we oriented our subsequent designs towards the IEEE 802.15.4 standard, a simplified, reduced power consumption, sensor network oriented, wireless protocol. Its implementation will be discussed in the next sections.

Section 3 will highlight the limitations of our previous systems and the general bottlenecks we are trying to overcome. Sections 4 and 5 will detail our proposed improvements. The 6th section will discuss experimental results.

[1] Author measured up to 30ms on SSP Bluetooth units.

3. GENERAL ARCHITECTURE

3.1 PHY and MAC implementation

The IEEE 802.15.4 is a standard which specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs) [12].

It operates mostly on the 2.4 GHz (ISM) band and features a on-air raw data rate of 250 kbps. Once the MAC layer has been implemented, the actual useable bandwidth for user data drops to 101 kbps [6].

One important bottleneck limiting the data rate is the communication between the sensor acquisition system and a wireless unit such as a XBee OEM module [13]. It is frequently achieved most with a UART which tends to be slower than the wireless data rate therefore adding to the overall transmission latency.

As an example, 6 sensors are sampled on 10 bits (2 bytes) and transmitted over a 115200 baud UART take 1.041 ms to reach a Xbee module.

This duration has to be compared with the maximum duration of the radio transmission [6] :

$$t_{transmit} = t_{worst\ case\ channel\ access} + t_{frame\ transmission} + t_{turn\ around} + t_{ack}$$

For the 12 byte payload above :

$$t_{transmit}\ (ms) = 2.368 + 0.576 + 0.192 + 0.352 = 3.488\ ms$$

Therefore, the additional transmission from the acquisition unit to the wireless module adds dramatically 30% of latency to the whole system.

3.2 Sensor sampling

Sampling analog sensors using the shared ADC of the microcontroller unit increases the used CPU, even if handled with interrupts. Most embedded sensor hardware cannot afford the proper analog front-end that would improve the multiplexer switching time because of the room it requires.

The obtained slew-rate relies the internal clock scheme and the sensor current sourcing. An average value of 120 µs acquisition time can be easily obtained with a maximum of 1 LSB of cross-talk between channels (16 MHz PIC microcontroller). Our 6 DoF sensor would require at least 720 µs of acquisition time using that sampling method.

3.3 User data protocol (OSI layers 5-6-7)

Formatting and translating sensor data is also a source of latency. To avoid adding latency, the system must have a coherent data encapsulation scheme and avoid packet translation to minimize impact on the system performance.

4. IMPROVED DESIGN

4.1 Hardware

Our previous design used a Xbee Zigbee module which stacks its protocol over 802.15.4. with no user access.

In order to virtually suppress the MCU to MAC/PHY latency discussed above and to author our own firmware and packet protocol, we opted for a microcontroller featuring 802.15.4 internally. A combination of hardware and software makes the data transmission between the user program and the MAC layer as fast as a RAM transfer. Data is further shifted out through the QPSK radio modem by the internal hardware.

We use a Jennic 5139R1 OEM module [14] that embeds a 16 MHz, 32 bit RISC microcontroller, 96 KB of RAM, a 802.15.4 MAC software and hardware stack associated to a 2.4 GHz radio as well as several handy peripherals to interface with sensors and external devices (I^2C and SPI digital interface, UART, GPIO, ADC, DAC and comparators).

The Jennic MCU is programmed in C language. Low level hardware access is eased with API functions while the 802.15.4 stack is proposed as software template. The module is powered by a 3.7V Li-Po battery cell and embeds its own charger.

4.2 Sensors

To reduce the ADC sampling scheme we use pre-digitized sensor read using the I^2C bus.

Our sensor node is composed of an Analog Device ADXL345 3D accelerometer and an Invensense ITG-3200 3D gyroscope. Each node can be extended with more sensors using either I^2C compliant digital sensors or optional accessory boards translating classic analog sensors to I^2C.

4.3 Communication with the host computer

A base station was developed as a WPAN network coordinator using also a Jennic MCU. The base station cannot really be considered as just a receiver since it achieves communication both ways, just like the sensor nodes.

Popular solutions such as Arduino use a serial port to send data to the computer, sometimes over a SLIP socket. While easy to implement on small MCU, serial links add to the latency of the system ; in order to keep the data path as fluid as possible and avoid further bottlenecks, we used ethernet to communicate with the computer, this time using a Wiznet812 100BASE-T module rather than our former 10BASE-T solution, therefore dividing the corresponding latency by a factor 10. Moreover, ethernet allows up to 100 m long data links to the computer.

Data is exchanged with the host computer using the OpensSoundControl (OSC) protocol over UDP, allowing easy data parsing as well as up-link configuration parameters that can be sent to the sensor nodes as discussed in section 3.4.

The Datagram contents is sent from the Jennic MCU to the Wiznet module via a 8 MHz SPI bus ensuring a inter-component high speed data exchange.

Finally, the base station is configured using a web server hosts and a parameter page accessible in a web browser.

4.4 Protocol and Services

Our goal was a generic sensor node capable of accepting sensor *accessories* as illustrated in figure 4. Specifically, this corresponds to extend the node with external daughter boards. There is no hardware dependency coded in the firmware aside the I^2C driver of the two 3D onboard sensors.

On top of this direct sensor access, we developed a service oriented protocol that allows extra peripherals to be discovered and read by the sensor node. The sensor node is designed as a data collection hub : the hardware dependencies are located in the sensor itself that then communicates with the hub over a high speed I^2C bus using an intermediate 16 bit PIC24F64GA004 microcontroller per accessory. This might also benefits from some local sensor processing (filtering, triggering, sample rate control).

This topology allows for the implementation of various sensor interfacing, such as analog sensors, piezo microphones, matrixed keypads or digitized SPI sensors. I^2C sensors can be used too since the PIC MCU has two I^2C ports. The low level I^2C driver is coded in the intermediate MCU and forwarded over our frame oriented I^2C exchange protocol within the sensor node.

Each accessory, as well as the onboard sensors can have several services which are proposed by the sensor node during the discovery phase. Most of the accessories we designed have

a streaming service with controllable sample rate but can also have high-level pre-processing such as Kalman filtering, onset detector or others algorithms for which an accurate sampling rate is mandatory for proper results. Interfaced as an accessory, continuous analog sensors can be turned locally turned into triggers without the need of a continuous streaming. As a result, more sensor nodes can be used simultaneously in the network without degrading the bandwidth.

Each service sends data frame containing the frame type tag (ints or floats). Each sensor node is identified by a hardware ID installed into the module FLASH using its serial port. The hardware ID is used in the OSC address scheme to route each module data on the host computer :

```
/<hardware ID>/<service #>/ data list
```

All radio communication, MAC addresses and higher level addressing scheme is transparent for the user. The base station WPAN coordinator handles node association and de-association on its own using our implementation of a simplified ARP table. The latter is periodically broadcasted to all nodes on the network to keep each node aware of the network population.

Finally, each radio frame receive a 1 byte packet number to detect short term data drops and the base station uses a sub-millisecond timer to date each packet exported in OSC.

5. IMPLEMENTATION

Since the sensor node is a scalable platform, we were able to use the same hardware base for both current projects.

For the Interlude project, the sensor node took the shape of the "MO" handheld unit accepting plug-in accessories as described in 4.4. The figure 1 show examples of accessories containing daughter boards as described in 4.4.

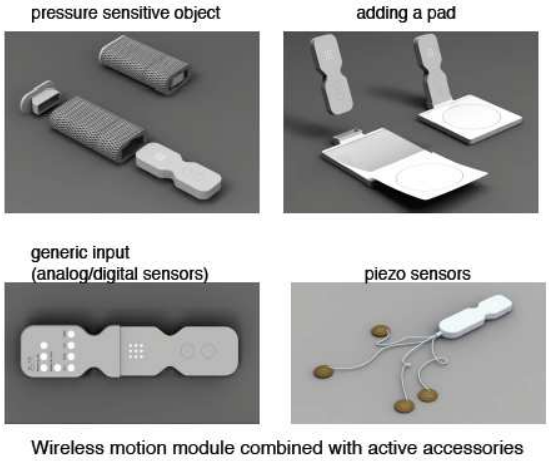


Figure 1. MO configuration array for the Interlude Project (design by NoDesign)

In its smallest form factor (mini-MO), the unit features only the onboard 3D sensors, no led array nor accessories and it becomes a 50x30x13 mm wearable unit rechargeable over a USB port.

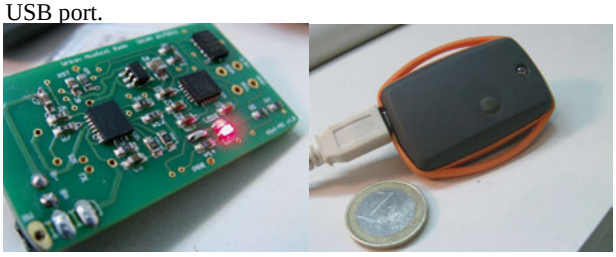


Figure 2. The Mini-MO configuration

6. SYSTEM EVALUATION

6.1 Internal timings

We measured the key acquisition timings using an oscilloscope and a GPIO of the Jennic MCU, setting the GPIO at the beginning of the function call and clearing it at the end of the process.

6.1.1 Data transfer to the MAC/PHY layers

Our final packet formatting is achieved by adding 8 bytes to the "useful" data payload : the frame delimiter, data type tags, message type, packet number and CRC.

For the same sensors' data payload used in the example in section 3.1 (12 bytes), the 802.15.4 frame transmitting the onboard sensors data takes 64 µs to be sent to the Jennic internal MAC stack. This is the most important improvement compared to a separated radio module communicating using a serial port (16 times faster).

6.1.2 Sensor data retrieval

With a speed of 400 kbps, the I^2C bus allows us to retrieve the 3D acceleration (10 bit sampled) and 3D angular speed (16 bit sampled) in 312 µs, API function calls included. This improves the sensor acquisition time by a 2.3 factor.

6.1.3 Base station latency

The build of the OSC packet and its data transfer by SPI also adds some latency. We measured the lag at two locations of the program using GPIOs. The base station processing adds 1.64 ms. This highlights again even a short, optimized OSC message takes some time to be assembled by embedded electronics [5].

6.2 Overall latency

We also measured the actual latency (best case) from the sensors themselves to the reception of the OSC packet in a computer using Max/MSP and an audio card.

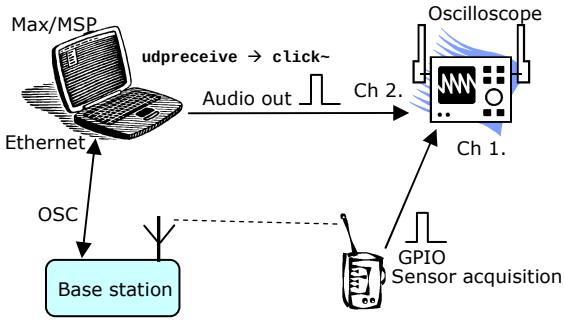


Figure 3. Latency measurement technique

We measure the lag between sensors acquisition and audio click and we kept the shortest duration as a reference.

Total actual latency = measured latency - audio latency

Total actual latency (best case) = 31.8 - 27.6 = 4.2 ms

Using the measurement of 4.1.3 we can conclude the OSC packet latency throughout the operating system and Max/MSP is 400 µs.

Table 1. Latency costs summary for different systems (ms + best case)

	Sensors	Radio	Base station	Host	Total
Xbee Serial	1.25	3.62	7.14	n.a.	12.01
Xbee API	0.8 (est.)	3.62	1.01	n.a.	5.43
Jennic	0.312	1.848	1.64	0.4	4.2

The best case measurements were obtained with no back-off wait from the CSMA/CA channel access. Hence the overall latency expected with our system is {4.2 ; 6.568 } ms.

Using the time stamp generated by the base station, data flow can be re-aligned and lost packets can be detected.

The variable part of the latency is essentially the radio transmission. We experimented a minimum transmission period of the on-board sensors streaming service of 3.2 ms. Up to 3 sensor nodes can be used simultaneously in continuous streaming while staying under the accepted 10 ms range. More nodes can be used with non continuous or asynchronous services.

6.3 Runtime

The Mini-MO version of the sensor node uses an average 52 mA. We use a 290 mAh PCB protected Lithium-Polymer battery pack conferring the device more than 5 hours of continuous use.

7. CONCLUSION

The paper presented how we improved certain aspects of wireless sensors data acquisition using the IEEE 802.15.4 standard. We showed that by using an integrated solution for the sensor node MCU and radio modem, as well as using digital sensors, the system can be 3 times faster than existing solutions using the same radio standard.

While we chose to have an extended radio packet protocol, raw data could be sent hence reducing radio and OSC packets building times resulting of further latency reduction.

The use of the I^2C bus allows a good scalability rather than relying on the number of ADC channels available on the sensor node MCU.

Upcoming work is the design of several sensor accessories for the MO handheld version to build an actual network population in order to evaluate the system performances with a larger number of participants (4 to 8).

The smaller implementation of the sensor node ("Mini-MO") will be used from now as our standard gesture capture unit for dance and augmented instrument projects.

8. ACKNOWLEDGEMENTS

This research is funded by the National Research Agency (ANR) and CapDigital (Interlude project ANR-08-CORD-010), and la Région Ile-de-France.

We would like to thanks the projects consortiums and particularly NoDesign and Da Fact for the design of the MO platform.

9. REFERENCES

[1] Aylward R., Paradiso J., Sensemble: A Wireless, Compact, Multi-User Sensor System for Interactive Dance. In *NIME* 2006.

[2] Bevilacqua, F., Guédy F., Schnell N., Fléty E., and Leroy N., Wireless sensor interface and gesture-follower for music pedagogy. In *NIME* 2007.

[3] Coniglio, M., Stoppiello D., The MIDI dancer. Troika Ranch Web Article - *http://www.troikaranch.org/pubs/ Movement_Research_Paper.pdf*

[4] Flety, E., The WiSe Box: a Multi-performer Wireless Sensor Interface using WiFi and OSC. In *NIME* 2005.

[5] Fraietta, A, Open Sound Control : Constraint and Limitations. In *NIME* 2008.

[6] Jennic. JN-AN-1035, Calculating 802.15.4 data rates. Application note. In *http://www.jennic.com/ files/support_files/JN-AN-1035%20Calculating%20802-15-4%20Data%20Rates-1v0.pdf*

[7] Malloch, J, Wanderley, M. , Sense/Stage - Low cost Open Source sensor infrastructure for live performance and interactive, real-time environments. In *ICMC 2010*.

[8] Mathews, M., The 1997 Mathews Radio-Baton & Improvisation Modes. In *ICMC* 1997.

[9] Rasamimanana, N. & Al., Frechin, J-L., Petrevski, U. Modular Musical Objects Towards Embodied Control Of Digital Music. In *TEI* 2011.

[10] Schmeder, A., Freed, A., Implementation and applications of Open Sound Control Timestamps. In *ICMC* 2008.

[11] Theremin, L., Method of and apparatus for the generation of sound. *US patent 1,661,058* - 5th of December 1925.

[12] Torresen, J., Renton E., Jensenius, A., Wireless Sensor Data Collection based on ZigBee Communication. In *NIME* 2010.

[13] http://www.digi.com

[14] http://www.jennic.com

[15] http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf

Snyderphonics Manta Controller, a Novel USB Touch-Controller

Jeff Snyder
Princeton University
310 Woolworth Center
Princeton, NJ 08544
jeff@scattershot.org

ABSTRACT
The Snyderphonics Manta controller is a USB touch controller for music and video. It features 48 capacitive touch sensors, arranged in a hexagonal grid, with bi-color LEDs that are programmable from the computer. The sensors send continuous data proportional to surface area touched, and a velocity-detection algorithm has been implemented to estimate attack velocity based on this touch data. In addition to these hexagonal sensors, the Manta has two high-dimension touch sliders (giving 12-bit values), and four assignable function buttons. In this paper, I outline the features of the controller, the available methods for communicating between the device and a computer, and some current uses for the controller.

Keywords
Snyderphonics**,** Manta, controller, USB, capacitive, touch, sensor, decoupled LED, hexagon, grid, touch slider, HID, portable, wood, live music, live video

1. INTRODUCTION
In early 2008, I invented the Manta to solve my own performance dilemmas, and to give myself a more expressive interface with my computer audio software. I soon found, however, that other composers and performers were interested in the capabilities of the device, so I redesigned the instrument to be possible to manufacture in small quantities and released it as a commercial product in May 2009. There are, as of my current writing, around 130 Manta users worldwide, and many of them use the controller for purposes I never originally imagined or intended.

2. DESIGN FEATURES
2.1 Sensor Layout
The most salient feature of the Manta is the hexagonal sensor lattice. There are six rows of eight sensors each, totaling 48 hexagonal sensors. Each sensor is capable of sending information about how much surface area is covered by the performer's finger, independent of the other sensors. If you use the sensors as each triggering separate "notes", then this can be seen as an implementation of polyphonic aftertouch. The data sent by each sensor is slightly less than 8 bits, since digital conversion is performed at 8-bit resolution, which is reduced by a built-in headroom[1]. In addition to the hexagonal sensors, there are two touch sliders that send centroid data at 12-bit resolution, and four assignable function buttons, which are the same technology as the hexagons but are visually distinct from them on the layout to simplify their use as user-defined special-purpose buttons. The sensor layout is fixed, but the data the sensors send is general and could be used for any purpose the user wishes. One could argue a significant disadvantage to a fixed sensor layout in an age when the interface trend is toward infinitely flexible touch-screen interfaces, but there are also several advantages to a fixed sensor layout in a musical instrument. For one, an unchanging physical distance between the sensors encourages the development of muscle memory on the instrument. Also, it allows for the use of subtle tactile feedback, which would be harder to implement on an interface like a touchscreen.

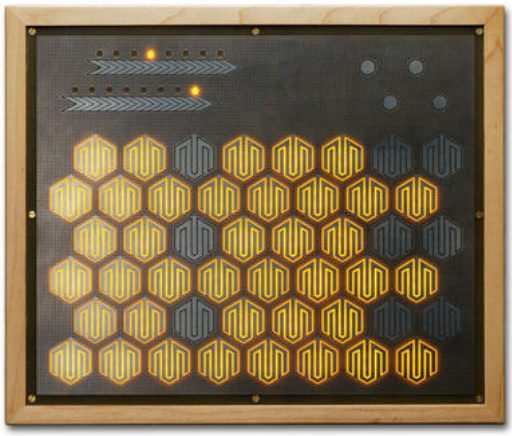


Figure 1: The Snyderphonics Manta controller

2.2 The Hexagonal Lattice
The main inspiration for the use of the hexagonal lattice pattern was an interest in the theoretical work of Ervin Wilson, whose microtonal keyboard designs are in turn inspired by the regularized keyboard designs of the 19th century, like those of Paul von Jankó or Robert Bosanquet[1][3], However, the limited number of available "keys" on the Manta, when compared to a design such as Bosanquet's, reduces the possibilities for redundant unison notes in a pitch layout, removing some of the advantages of Bosanquet's layout, such as unbroken identical scale "shapes" regardless of key center. Similar developments that focus on button instruments like the

[1] This headroom is necessary to compensate for sensor drift (due to temperature and EM noise), which is monitored and compensated for by the firmware.

accordion have adapted the Jankó/Bosanquet idea, but adapted it to be more useful with a limited number of buttons, by removing the vertical unisons and replacing them with octaves. For instance, the Wicki-Hayden system[2] implements this change, while also rearranging the pitch assignments between rows to put fourths and fifths nearby, as opposed to the Jankó design, which places semitones in this position. Therefore, if one intends to use the Manta as a keyboard interface with pitches assigned to the hexagons, accordion layouts are more appropriate to the limited number of sensors on a Manta, and the 48-sensor arrangement allows for a simple implementation of the layout shown in Wicki's patent[3], with the omission of three repeated pitches[4]. Nevertheless, any Wicki-Hayden keyboard cannot achieve the advantage of truly transposable scale shapes, avoiding so called edge-effects, without at least 100 sensors[5], so one of the primary advantages of regularized keyboards is compromised for the sake of size and portability. Since my intentions for the instrument were not necessarily to always use the sensors as "keys" assigned to particular pitches, this compromise was a design choice and involved a tradeoff between sensor size, cost of sensing components, and overall device footprint.

Considered as a more general concept, the hexagonal grid affords the user three degrees of close relationships between directly adjacent sensors, which are, in the case of the Manta, horizontal and the two diagonals. These adjacencies can be inspiring for avoiding more standard rectangular grid control mappings. There are, of course, several similar applications of the hexagonal lattice to music controllers, such as those designed by C-Thru Music[6], Thumtronics[7], Starr Labs[8], Cortex Design[9], and Opal[10], and all of these are building upon either the Jankó layout or some variant of the Euler/Reimann Tonnetz[11]. However, to my knowledge, the Manta is the only commercial touch controller that combines this type of layout with the benefits of capacitive touch sensing.

2.3 LED feedback

Each hexagon and function button can be backlit in either red or amber, and this functionality can be computer-controlled. By default, the amber LED behind a sensor turns on when the sensor is touched, but this direct coupling can be deactivated with a command from the computer, after which the LEDs are completely under computer control. This functionality opens up several possibilities for more complex or context-specific visual feedback, and its primary inspiration was the Monome controller, designed by Brian Crabtree and Kelli Cain[12].

2.4 Capacitive Touch Sensing

Capacitive touch sensing is the basic sensing apparatus for the Manta. I was particularly inspired by the 100-series touch controllers designed by Donald Buchla[13], which I had the opportunity to use while studying for my doctorate at the Columbia University Computer Music Center. While these touch controllers were not polyphonic in the same way the Manta is, I found the control they afforded to the user to be very satisfying, and I decided that a surface-area-based capacitive sensing design would give me the expressive capabilities I wanted as a performer. Capacitive sensing in electronic musical instruments goes back at least to the Theremin, and has been used in musical instruments throughout the 20th Century. Examples of effective musical instruments using this technology include the Trautonium by Friedrich Trautwein, the left-hand controls of the Electronic Sackbut by Hugh Le Caine, the Multiply-Touch-Sensitive keyboard by Bob Moog and Thomas Rhea, the Wasp by EDP, the Synthi-AKS by EMS, and the Sal-Mar Construction by Sal Martirano, and the EVI and EWI by Nyle Steiner.[14] The technology has seen a recent resurgence in other markets due to the current trend of capacitive touch screens and buttons implemented by portable devices like the iPod and iPhone[15]. I find it to be a very useful sensing method for musical purposes, although I think the typical approach for modern devices of placing a glass or plastic sheet above the sensors reduces the tactile feedback to the point where the usability of the approach is significantly diminished. I designed the Manta to give the users direct contact with the metal traces of the sensors, which are etched onto a circuitboard laminate so that there is some amount of tactile feedback. This also makes sliding, glissando gestures more satisfying to the user, since the friction of the surface is less than that of glass or most plastics, such as that used on the iPad.

2.5 Velocity Detection

Early in the development of the Manta, I found that while the continuous sensor data the Manta outputs was very inspiring and suggested many expressive uses, I was also often interested in getting standard note-on and note-off data, with velocity. This is the information usually conveyed in the keyboard controller paradigm. On a standard electronic music keyboard, when you press a key, the keyboard reports which key you pressed, and how fast that key went down. This value is called velocity, and is usually mapped to amplitude of the resulting sound. When you release the key, the information about which key was released is sent. Note-on and note-off are trivial to implement in a capacitive touch-sensing system by simply determining a threshold of capacitance measurement and reporting when that threshold is crossed, possibly with some hysteresis and de-bouncing. However, "velocity" data is much harder to determine on an interface with no moving parts. It is impossible to measure the time it takes for a key to go down if the key doesn't move.

After much experimentation and collaboration with Angie Hugeback, a statistician and postdoctoral researcher at the University of Washington, I found a technique that produces a reasonably reliable velocity data based on information gathered from two successive samples above the "on" threshold. The technique involved training on example data, and using machine learning to generate an algorithm that could be applied

[2] The Wicki-Hayden system is so named because it was originally discovered by Kaspar Wicki and patented in 1896, and later independently discovered by Brian Hayden and patented in 1986

[3] Swiss patent Nr. 13329

[4] The Wicki patent diagram is for a 51-note bandoneon.

[5] This limitation is discussed in a paper on the Wicki-Hayden keyboard by Robert Gaskins at http://www.concertina.com/gaskins/wicki/index.htm

[6] http://www.c-thru-music.com/

[7] http://www.thummer.com/

[8] http://www.starrlabs.com/

[9] http://www.cortex-design.com/projects_terp1.htm

[10] http://www.theshapeofmusic.com/

[11] http://en.wikipedia.org/wiki/Tonnetz

[12] http://monome.org/

[13] http://www.buchla.com/

[14] A good overview of many of these instruments is available at http://web.media.mit.edu/~joep/SpectrumWeb/SpectrumX.html

[15] http://www.apple.com/

to the continuous data stream. Because the Manta was designed to be a commercial product, and cost is therefore a factor, the design I chose limits me to a minimum scan rate of around 6-8ms for each sensor.[16] This means that by the time two successive samples above the threshold have been collected, the elapsed time is just past the threshold of human latency perception. This meant that two samples is all I can use without the velocity algorithm feeling too slow, and the 6-8ms scan rate also guarantees that most of the sensor information happening within the very fast action of the initial key touch has been lost. Nevertheless, we were able to produce a surprisingly successful algorithm, which we implement in the host computer software rather than the Manta hardware to avoid the additional computation time it would add to the MCU loop. The addition of this velocity-detection algorithm allows the host computer to output both traditional note-on/note-off with velocity and the more unusual polyphonic continuous data simultaneously; the user can choose which data to use and which data to ignore, or combine the two data streams for a note-on with polyphonic aftertouch effect.

3. MANTA COMMUNICATION

There are currently three ways to communicate with the Manta on a computer – the Manta Max object, MantaCocoa, and libManta. I am in the process of developing the MantaMate, a dedicated hardware device that will communicate with the Manta without the use of a multimedia computer.

3.1 The Manta Max Object

The Manta is a USB controller, with a built in mini-B female connector. Since it is mostly HID class compliant, the built-in HID drivers on the Windows and Mac OS X operating systems correctly identify it[17]. However, because it is a vendor-specific HID device, another layer of software is needed to present the data to other programs that may want to use it, such as Max/MSP or Abelton Live. There are currently three ways to access the data from the Manta, and to send data to the Manta. The first way is via the [manta] Max/MSP object development by Brad Garton and myself. The [hi] object was not usable for this purpose since it does not include the ability to send output reports (which are needed to control the LEDs), so we developed a custom object, written in C, that pulls the data from the HID driver and presents it to Max/MSP, and also allows for Max/MSP users to send data to the Manta. Additionally, it makes it possible to route the data from the Manta to other applications using the Max/MSP free runtime environment and a patch that sends the Manta data over MIDI or OSC. The Manta Max object is available from the Snyderphonics website[18]. Damon Holzborn has added to this work by creating a Max For Live patch that utilizes the [manta] object and easily interfaces the Manta with Abelton Live.

3.2 MantaCocoa, The Manta OSC router

Jan Trützschler von Falkenstein has written a standalone program in Cocoa for Mac OS X that presents the data from the Manta as OSC messages, and receives OSC messages to control the LEDs and various operation modes of the Manta. This program makes the use of the Manta easier for Mac users who don't wish to use Max/MSP. MantaCocoa is especially popular among SuperCollider users who perform with the Manta. MantaCocoa is available from Jan Trützschler's website[19].

3.3 libManta

Spencer Russell has released a beta version of libManta, a C++ library to present a simple, consistent, cross-platform API to those programming applications for the Manta. It is based on libUSB, and takes care of the asynchronous polling of the USB driver and the formatting of the bit packets that the Manta understands.

libManta is available at http://gitorious.org/libmanta and is released under the GNU Public License. Spencer has also been working on a Flext object for the manta, which works on PD as well as Max/MSP, and includes Linux support, as well as an open-source cross-platform OSC router for the Manta. Christopher Jacoby has recently joined the development team and is nearing a beta release of a standalone Manta MIDI router for Mac and Windows built on the libManta library.

3.4 The MantaMate

I am currently working on the hardware for a new device that will allow the Manta to more easily interface with voltage-controlled analog synthesizers. It is basically an embedded USB host, with four 16-bit DACs, and eight 12-bit DACs. I call the device the MantaMate, and it is currently in the prototyping stages. It is conceived with the goals of enabling 4-note polyphony for a wide range of voltage controlled synthesizers, communicate with the Manta without the use of a computer, have sufficient accuracy and resolution for the implementation of unusual tuning systems on analog synthesizers, and support both OSC over Ethernet and MIDI.

I intend to release the MantaMate as a commercial product once the prototype has been fully developed and tested. I conceive of the MantaMate as not just an interface for the Manta, but also as a general-purpose format converter for musical communication – allowing the conversion between OSC, USB-HID, USB-MIDI, MIDI, and CV standards. It is not, however, an embedded computer, as it is not designed to be able to run an operating system[20].

4. USES FOR THE MANTA

Manta users have found several ways to put the capabilities of the controller to use. I'll outline a few of them here.

4.1 Microtonal Keyboard

In my own music, I have primarily used the Manta as a microtonal keyboard. I have developed what I consider the

[16] I wanted to avoid any solution that uses the audio interface for the computer to handle the data stream from the device, and I also wanted to avoid bogging down the host computer with 54 dimensions of fast data (the 48 hexagons, the 4 function buttons, and the 2 sliders). This rules out approaches like those implemented in David Wessell's 2-D touch design [5], or Madrona Labs device [2]. However, I believe in the final design by Madrona Labs, they have integrated the DSP into the hardware. When designing the Manta, I chose not to use a dedicated DSP chip, so my scan possibilities are more limited. The 6-8ms latency includes processor overhead and USB transfer rate.

[17] When programming the original Manta firmware, I used a 16-byte USB output report. This is actually outside the HID spec (4-byte maximum is specified), but I didn't notice the problem because both Mac OS X and Windows ignored the issue. Further testing shows that Linux complains, but the issue is easily sidestepped by using libUSB instead of libHID, accessing the raw USB reports. Spencer Russell implemented this workaround.

[18] http://www.snyderphonics.com

[19] http://falkenst.com/

[20] It is based on at Atmel AVR32 series of MCUs, not an ARM architecture. Therefore, it's not possible to, for instance, run PD patches or Chuck programs on it.

"concert version" of the instrument, which consists of two Mantas side-by-side on top of a custom-built wooden resonator. My standard software patch considers each hexagonal sensor a separate note, and maps the continuous data from that sensor as the amplitude of that note. This allows the performer to fade in each note of a chord independently, and control the relative volumes of the pitches with careful precision. It also allows for a very expressive tremolo effect. Putting the two Mantas side-by-side achieves a 96-note playing surface, without sacrificing the portability much, since they stack up to around 0.7" thick and roughly the length and width of a 15-inch laptop during transport. The wooden resonator has an electromagnet attached to a spruce "top", which is driven by an amplified signal from a computer running a Max/MSP patch. This gives the "concert Manta" a characteristic sound by acoustically filtering its digitally synthesized voice. Usually, I write music to be played on the Manta by other people, combined with an ensemble of other instruments I have designed. I see it as the keyboard family in my invented orchestra. The hexagonal lattice helps to avoid the equal-tempered expectations performers have about standard piano keyboards. I describe the microtonal system I use on the Manta in detail in my doctoral dissertation, Exploration of an Adaptable Just Intonation System[4].

Other users, such as composer Stephen James Taylor[21], also find the Manta appropriate for this purpose. It's also, of course usable as a controller for standard 12-tone equal temperament, in which case the unusual keypad layout can serve to avoid stereotyped keyboard habits.

4.2 Interface for Live Processing

Sam Pluta[22], an NYC-based composer and electronics performer, was an early adopter of the Manta, and was extremely helpful in early beta development of the hardware. He performs live, improvised electronic music on the Manta. He wrote his own custom software in SuperCollider, which allows him to record, manipulate, and process audio coming into his computer from a microphone. He usually performs in combination with acoustic players, such as the trumpet player Peter Evans, grabbing, stretching, distorting, and otherwise transforming their performances into strange and otherworldly textures[23]. He often treats each sensor as a control for a particular function, sometime using the continuous data, sometimes the velocity data, and sometimes just the simple on/off data. He finds the continuous values from the sensors to be extremely useful to, in his words, "really get your fingers on the data in your computer"[24]. As a gigging musician in NYC, where one generally needs to get to a gig via subway, he finds the compactness and portability of the manta to be especially suited to his needs.

Other users have also applied the Manta to a similar live-processing purpose, including Christopher Jon, the keyboardist and synthesist for the band Android Lust[25], who uses the Manta live to process the voice of the lead singer. He uses the centroid-detection mode built into the [manta] Max/MSP object[26] to control DSP effects applied to the singer's microphone input by sliding his hand around the hexagon grid.

[21] http://www.stephenjamestaylor.com/

[22] http://www.sampluta.com/

[23] You can hear some of these improvisations on the 2011 recording "Sum and Difference", from Carrier Records. http://carrierrecords.com/

[24] From personal correspondence with Sam Pluta

[25] http://www.androidlust.com/

[26] The centroid detection mode was developed by R. Luke Dubois, and finds a centroid when a large area of the hexagonal grid is covered by the hand, such as when the user applies two or three fingers in close proximity. The discrete layout of the sensors makes this sensing method much less accurate than a similar function on a touch-screen, but it is still musically useable as a low-resolution x-y controller.

[27] http://www.victoradan.net/

[28] http://vimeo.com/4611451

[29] http://music.columbia.edu/~daniglesia/

4.3 Unusual Uses for the Manta

I have a band with the composer Victor Adan[27], in which we each use a Manta to control old pen-plotters we have bought on E-bay. The band is called the Draftmasters[28], and we perform music by sending commands to the plotters and amplifying the motors that move the pen with electromagnetic pickups. Each piece by the Draftmasters uses the hexagonal layout differently – in some cases many of the sensors are X/Y coordinates for the pen to move to, in others they send different pen speed commands to change the frequencies the motors generate. We collect the Manta data in Pd, send it to a python script, and then output it to the plotters as serial commands in HPGL.

Dan Iglesia[29], a composer and video artist living and working in NYC, uses the Manta to control live 3D video, generated with OpenGL in realtime. He wrote the custom software he uses in Jitter and simultaneously controls the audio and the video from the Manta controller live.

5. Future Development

In the future, I hope to finish the MantaMate hardware to make interaction with both newer and older analog synthesizers more simple, as well as the control of MIDI hardware and networked devices. Also, I believe that Spencer Russell's libManta, when fully released, will make development of host computer software for the Manta much easier. I consider the Manta hardware itself to be stable and unlikely to undergo significant changes, or at least I aim to make any future changes backward-compatible. Further development is mostly focused on creating software that makes the Manta easier to use and more robust for software applications.

6. ACKNOWLEDGMENTS

My thanks to Brad Garton, Sam Pluta, Jan Trützschler von Falkenstein, Spencer Russell, Angie Hugeback, Christopher Jacoby, and Damon Holzborn.

7. REFERENCES

[1] Bosanquet, R.H. *An Elementary Treatise on Musical Intervals and Temperament.* Diapason Press, London, 1876.

[2] Jones, R., Driessen, P., Schloss, A., Tzanetakis, G., A Force-Sensitive Surface for Intimate Control. In *Proceedings of New Interfaces for Musical Expression* (NIME). 2009.

[3] Keislar, D. History and Principles of Microtonal Keyboards. *Computer Music Journal* 11, 1 (Spring 1987), 18-28.

[4] Snyder, J. *Exploration of an Adaptable Just Intonation System*. D.M.A. Thesis, Columbia University, New York City, NY, 2011.

[5] Wessel, D., Avizienis, R., Freed, A. and Wright, M., A Force Sensitive Multi-touch Array Supporting Multiple 2-D Musical Control Structures. *Proceedings of New Interfaces for Musical Expression* (NIME), 2007, 41-45.

On Movement, Structure and Abstraction in Generative Audiovisual Improvisation

William Hsu
San Francisco State University
1600 Holloway Avenue
San Francisco CA 94132, USA
whsu@sfsu.edu

ABSTRACT

This paper overviews audiovisual performance systems that form the basis for my recent collaborations with improvising musicians. Simulations of natural processes, such as fluid dynamics and flocking, provide the foundations for "organic"-looking movement and evolution of abstract visual components. In addition, visual components can morph between abstract non-referential configurations and pre-defined images, symbols or shapes. High-level behavioral characteristics of the visual components are influenced by real-time gestural or audio input; each system constitutes a responsive environment that participating musicians interact with during a performance.

Keywords

Improvisation, interactive, generative, animation, audio-visual

1. MOTIVATIONS

In the last few years, I have been working on a series of audio-visual pieces for performance with improvising musicians. Each piece is an interactive animation environment that responds to gestural input and real-time audio. The animations are projected on stage with the musicians. The video becomes an additional component in the interaction between the musicians during the performance. The musicians cannot directly control the details of the animations; they are improvising with each other, and with the animations. Each system is primarily generative; the initial specification of a minimal amount of source material (usually a few images or shapes) will result in a wide range of dynamically evolving structures and behavior for different performances. Hence, my work takes a different approach from systems such as VERSUM [1], which is primarily an audiovisual sequencer with powerful spatialization capabilities.

While the concept and "look" of my pieces can vary significantly, there are some unifying themes that inform much of my work in this area. As a performer, system designer, and listener, I have noted the associations we often make between abstract gestures and forms, and natural phenomena. I have tried to build performance environments that capture some of these associations.

My animation environments are designed primarily for working with free improvisers. Hence, I work mostly with abstract visual components, such as particle clusters, lines and curves. Gestural and timbral cross-referencing between sound and visuals evokes the tactile, nuanced, timbrally rich gestures that I enjoy in improvised music. The complex behavior observed in natural processes, such as fluid dynamics and flocking behavior, seemed promising to me for evoking complex and dynamic gestures. A number of my pieces are built around simulations of such processes. The resulting visual phenomena evolve in a complex and highly detailed manner, comparable with the gestures of free improvisers.

While the basic components of my animation environments are abstract particles and shapes, I am interested in setting up tensions between abstract elements and configurations that reference or evoke concrete objects or symbols. While a specific visual stimulus can fundamentally be interpreted in multiple ways, the human visual system tends to prefer one clear interpretation. One of my goals is to encourage situations where this interpretation is highly unstable, shifting from an unstructured configuration such as a pseudo-random point cloud, to a well-defined image with a fairly unambiguous interpretation, such as a human skull. In all the environments and pieces described in this paper, the abstract elements are able to coalesce into well-defined images, patterns or symbols, and eventually scatter into non-referential configurations. These evolutions in structure occur in the context of the simulated process that is the basis for each environment; they can be influenced by gestural input or real-time audio.

In this paper, I will describe two groups of pieces that incorporate improvising musicians and interactive animation. These are the *Interstices* pieces, which are based on particles in a fluid system, and the *Flayed/Flock* pieces, which are based on flocking simulations. I will focus on 1) using simulations of natural processes, such as fluid dynamics and flocking behavior, to achieve an overall framework for generating motion of visual components that evoke natural processes, 2) adapting the simulations to incorporate referential visuals, and handling transitions between abstract elements and images, and 3) interaction with gestural input and real-time audio in the context of live performance. Versions of these pieces have been performed at Sound and Music Computing 2009 (Porto), Steim (Amsterdam), and other venues in the United States and Germany.

2. PAINTING WITH PARTICLES

The *Interstices* group of pieces are based on the motion of particles in a fluid system. My starting point for the fluid simulation is Glen Murphy's Fluid code [7]. Murphy's code was also used in his *Fluid Bodies* installation [8]; viewers interact with the fluid system through a camera and projection setup. Viewers' movements cause changes in particle brightness and density, and "reflections" of the viewers form in the particle system.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2.1 Motion Generation

In the *Interstices* pieces, a system of up to hundreds of thousands of particles is manipulated with gestural controllers, such as a graphics tablet or a multi-touch device. Synthetic components such as attractors, repulsors and large tidal generators can be dropped into the system and set in motion. In addition, images (pre-loaded or captured in real-time) can be placed in the particle system, to be scattered apart by the simulated fluid movement; particles may also swirl and coalesce into images. The end-result resembles somewhat asymmetric, constantly morphing versions of the Rorschach inkblots used in psychological evaluations; the goal is to open up a wide range of visual associations.

My reworking of Murphy's code mostly involved increasing its efficiency to support large numbers of particles and a fine-grain 800 x 600 simulation grid; I found that the latter especially enhanced the highly-detailed look of the animation that I was aiming for. To enhance performance, the simulation grid is subdivided into 32 x 32 tiles; fluid velocity and pressure updates are skipped for tiles that contain very few particles.

In addition, I built a number of classes to implement high-level behavioral components in the fluid system. *Attractors* and *repulsors* are simply centers of positive or negative gravity in the fluid system, that influence the motion of the particles. These components themselves can be set in motion within the system. *Tidal generators* correspond to large sweeping gestures that influence the motion of particles in a large area; a *scheduler* object manages the aleatoric generation of sequences of actions to create tidal currents in the fluid system. With my customization and tuning of fluid simulation, these currents result in large "splashing" gestures that evoke breaking waves or painterly splatters. Such gestures can evolve for extended periods of time, through multiple shape configurations, before finally damping out. Interactions between tidal currents and attractors are also complex and unpredictable. A number of examples can be seen in the video clips at http://userwww.sfsu.edu/~whsu/PSHIVA. Figure 1 shows snapshots of the evolution of the particle system, over about 16 seconds, under the influence of a single tidal current, from a section in my video *A Way*. The ring-like shape is a final state that the system sometimes settles into, after many frames of chaotic behavior.

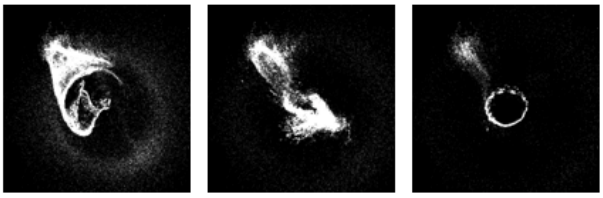


Figure 1: Examples of particle configurations as a result of tidal currents

2.2 Incorporating Pre-defined Structures

The particle system is also able to coalesce from a pseudo-random spatial distribution to an image or other well-defined referential visual structure, which I will call a *morph target*. A morph target might be pre-loaded, or captured from real-time camera input. Morph targets can be chosen and positioned dynamically during a performance.

In installations such as Murphy's *Fluid Bodies* [8], the shapes of viewers cause changes in particle brightness and density; concrete shapes are introduced to pseudo-random particle clusters by essentially fading in the shapes. This works reasonably well in some contexts; however, I felt that "fading in" an image results in an effect that clashed with the overall look of my particle-based pieces. I felt that having pre-existing particles move and collectively coalesce into an image was more compatible with the underlying fluid-driven motion.

To enable transition from an unstructured distribution to a morph target, the list of particles close to a morph target are simply mapped in a straightforward manner to the pixels in the morph target. The necessary linear trajectories are calculated for each particle to reach its corresponding pixel in the morph target. Over the next few seconds, each particle involved in the morphing activity would follow its pre-defined trajectory; when all such particles have traversed their trajectories, they will have coalesced into the morph target.

This is a very simple but efficient approach to handling the transition from unstructured particle clusters to images. Because of the simple mapping of particle to pixel, unnatural motion artifacts are possible; however, these tend to be obscured by the complex particle motion that is usually present. There are often over 100,000 particles in the system, so more sophisticated optimization techniques to minimize motion artifacts were too compute-intensive for our current development platform (an Intel Core 2 Duo with a low-end graphics processor); we do plan to explore other mapping strategies in the future.

Once a particle cluster has coalesced into a morph target, it is simple to let the underlying fluid simulation take over motion control for the particles. The fluid-based motion usually breaks up the morph target, and the particles return to a pseudo-random unstructured state. Figure 2 shows snapshots of a particle cluster coalescing into an image (a skull).

Figure 2: Particle cluster coalesces into a skull

2.3 Interaction with Gestural or Audio Input

I have tried several strategies for managing the audio environment for *Interstices*. In solo performances, with a single performer manipulating the particle system and audio generation simultaneously, my primary concern is to enable direct and detailed control of the particle system through gestural controllers. Hence, the performer's gestures correspond directly to events in the fluid simulation, "stirring" the underlying fluid to move particle clusters, placing or removing attractors or repulsors, guiding the formation of tidal currents, triggering the coalescing of particles into morph targets, etc. The same physical gestures are "interpreted" and loosely mapped to generative sonic gestures; there are several high-level options for the interpretation and mapping, which can be chosen by the performer to build contrasting sections of a piece.

Sonic gestures are synthesized by specifying high-level sound synthesis parameters such as duration, loudness, brightness, amplitude modulation etc. In a particular section, large physical gestures may result in loud, bright sonic gestures of long duration; in another section, sonic gestures may be restricted to shorter durations with very low brightness.

In addition, the particle system can respond to real-time audio

descriptors. For audio-reactive performances, I have used a customized version of Jehan's analyzer~ [4], and the Zsa descriptors [5] to generate audio descriptors. Estimates of activity level or timbral characteristics are extracted in real-time, and communicated via Open Sound Control to the animation environment. My preference is to avoid straightforward mappings of audio to animation parameters, such as brightness to position, etc, in favor of more open "interpretations" with multiple degrees of freedom. For example, the onset and continuation of a slow and loud sonic gesture may trigger a large tidal current in the animation; if the roughness of a sonic gesture is maintained above a threshold for a minimum time, a particle cluster will be triggered to coalesce into an image.

2.4 Implementation and Performances

The animation components of *Interstices* were developed in the Processing environment (http://www.processing.org). The sound analysis/synthesis component is a Max/MSP patch (http://www.cycling74.com). Gestural input from a tablet, touch screen or camera is captured and interpreted by Processing components, and sent to the sound synthesis components via Open Sound Control messages. Audio descriptors and other information captured by the Max/MSP audio analyzers is also communicated to the animation via Open Sound Control messages.

The first installment of the *Interstices* series was premiered at Sound and Music Computing 2009 in Porto. Subsequent performances have been at Steim (Amsterdam) in 2010, and at various venues in San Francisco and Germany, with musicians such as John Butcher, Chris Heenan, Gino Robair, Moe Staiano, and Birgit Ulher (using Ulher's drawings as morph targets in one section). A proposal for *Interstices AP*, a solo version with multitouch controller, has been submitted to NIME 2011.

3. FLOCKING FILAMENTS

Swarming or flocking has been widely observed in the collective behavior of migratory birds, ant colonies, schools of fish, etc. Flocking simulations have been used in generative art and music. Most of these projects simulate relatively small flocks of tens to a few hundred agents. [2] and [3] discuss the use of flocking/swarming agents to generate music.

My initial experience with flocking systems in the context of music improvisation was at the Live Algorithms for Music workshop in August 2009 in London. Tim Blackwell [3], Tom Mudd and I set up a chain of systems that improvised with percussionist Eddie Prevost. Prevost's live sound is analyzed, and descriptors are generated and distributed through software by Sebastian Lexer and Ollie Bown. The descriptors map to movement parameters of Blackwell's flocking simulation. My software module detected the presence, position and size of clusters in the flock; these were mapped to timbral space parameters in Mudd's software synthesizer.

Rowe and Singer's *A Flock of Words* [9] is a flocking animation of 10-30 words; flocking parameters are influenced by the performance of a chamber ensemble. My subsequent *Flayed/Flock* pieces utilize one or more flocks of thousands of particles that appear to "draw" abstract scribbles in space. In response to real-time audio, the flock formations evolve, and flocks are able to coalesce into well-defined shapes and symbols. To trace well-defined curves and patterns, dense flocks of thousands of particles are necessary; great attention must be paid to computational efficiency.

3.1 Motion Generation

Subsequently, I built a piece based on flocking, initially for use with Birgit Ulher and Gino Robair in San Francisco in February 2010. My starting point for the flocking simulation was Kyle McDonald's implementation [6], which simulates a single flocking population of particles. A particle moves through space based on Perlin noise; its motion is the result of forces affected by high-level parameters such as *speed* (an overall scaling factor for the velocity of each particle), *neighborhood* (essentially the extent to which a particle is influenced by nearby particles), and *spread* (the strength of an attractive force toward the centroid of the flock). I rewrote McDonald's code to support multiple flocks, improve its framerate significantly, and added the ability for the flock to coalesce into pre-defined patterns. Figure 3 shows two examples of curves traced by my flocking implementation.

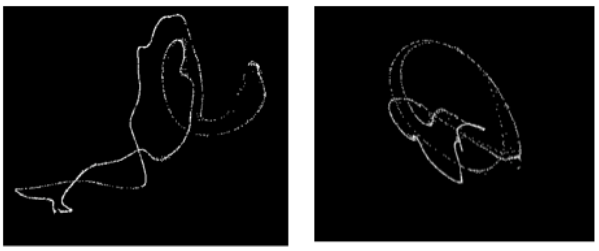


Figure 3: Examples of curves traced by simulated flocks

3.2 Incorporating Pre-defined Structures

The complex looping curves and lines traced by simulated flocks seemed to me to be a good match with simple referential shapes or symbols, such as circles, crescents or stars. For my current implementation, I restricted my symbols to relatively simple closed shapes. Black-and-white *masks* of the shapes, with one color assigned to the inside and one to the outside, are stored in image files that are loaded during performance.

As in the *Interstices* particle-based pieces in Section 2, I opted for having the flock coalesce collectively into well-defined target shapes, rather than resorting any fade-in effects. Again, the decision-making process for instigating the movement toward the referential forms must be highly efficient. To coax a flock into a pre-defined mask, I used the following algorithm:

1) determine the centroid of the flock (this is already part of the flocking simulation)

2) center the mask at the flock centroid

3) if a particle is outside the mask, it experiences an attractive force towards the centroid

4) if a particle is inside the mask, it experiences a repulsive force away from the centroid

This set of simple and efficient rules is sufficient to push particles to settle near the outline/boundary of the mask. For complex asymmetrical shapes, it is also possible to manually place multiple attractors within the shape, one for each section. However, the particles are often distributed in a very uneven manner around the outline. Hence, part of the outline might be clearly visible, having attracted many particles, but a significant part of the outline may be missing, having attracted few or no particles at all.

To even out the distribution of the particles around the boundary, I was inspired by a suggestion from Kazunori Okada (Okada, personal communication), resulting in a fifth rule:

5) if a particle is on the boundary of the mask, it will move from a region of high particle density to one of low particle density

With the last rule in place, the flock reliably settled into a number of pre-defined test patterns, distributing evenly around the outline of each pattern. Figure 4 shows snapshots of a flock, initially in a non-referential configuration, then slowly transitioning into a crescent shape. More examples of these transitions can be seen at http://userwww.sfsu.edu/~whsu/ARFlock.

Once the flock has settled into a pre-defined image or pattern, simply removing the constraints of the pattern will cause the flock to return to its previous abstract flocking behavior. (This can also be seen for example as part of McDonald's Janus Machine installation.)

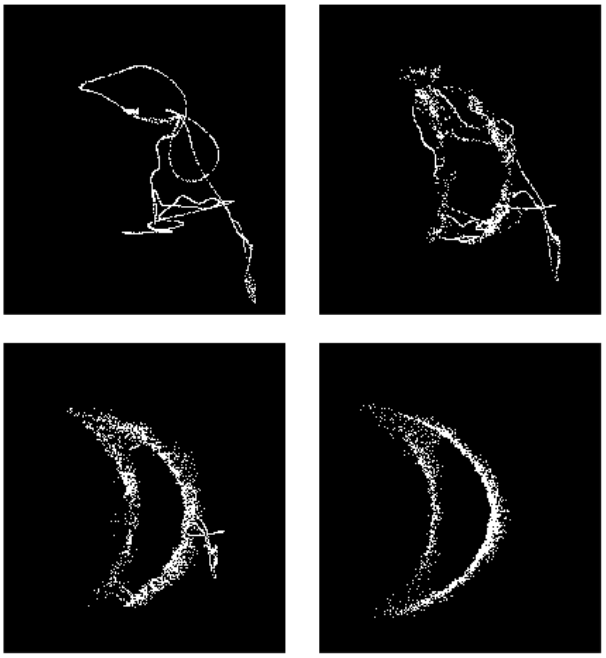


Figure 4: Transition of simulated flock from non-referential configuration to a pre-defined crescent shape.

3.3 Interaction with Gestural or Audio Input

In my design for flock's interaction with real-time audio, I wanted to avoid overly obvious mappings of audio parameters to spatial parameters in the flock's movement. My intention is to allow the flocking simulation to evolve based on its own rules, with the audio influencing high-level behavioral trends only. In earlier versions, I have again used simple activity measures for the audio, based on spectral flux [5]. Greater activity would increase the number of particles in the population, the speed of the population, and encourage the formation of coherent lines (by varying the *neighborhood* parameter); lower activity levels would decrease the number of particles and their speed, and encourage the flock to disperse in a random-seeming fashion. This can be clearly observed in the video clips available online. The details of the flocking behavior still evolve in a complex and unpredictable manner, but the high-level trends make clear references to the real-time audio.

For a flock to coalesce into pre-determined patterns and symbols, longer time intervals are required than are served by moment-to-moment monitoring of real-time audio descriptors. One approach I have explored uses an activity measure over a larger time window; as a musician remains active over a period of time, the analogy is to "energy" building up in a system. When a threshold is crossed, the flock undergoes a process of coalescing into a target pattern. As the activity level decreases over a period (i.e., energy levels dissipate), the particles abandon their previous target pattern, and return to the basic flocking behavior.

3.4 Implementation and Performances

Software for the *Flayed/Flock* pieces is structured in a similar manner as the *Interstices* pieces. Visual components are in Processing, audio components in Max/MSP, and gestural information and audio descriptors are exchanged via Open Sound Control.

Flayed/Flock has been performed with a number of free improvisers with varied insruments and approaches, such as its premiere at Artists Television Access gallery in San Francisco in February 2010, with Gino Robair (percussion, electronics), Birgit Ulher (trumpet) and myself (electronics), and at Steim (Amsterdam) in May 2010, with Gareth Davis (bass clarinet), Anne Laberge (flute) and myself. A proposal for *Flayed/Flock*, in collaboration with Oslo-based musicians Håvard Skaset and Guro Skumsnes Moe, has been submitted to NIME 2011.

4. SUMMARY

I have described the concepts and technology behind some audiovisual performance systems that I have built in the last few years. These were designed for my work with musicians, largely in a non-idiomatic free improvisation context. The generative visuals constitute complex, highly detailed gestures and textures, with unstable forms that encourage constantly shifting viewer interpretations. The systems have been used in a number of solo and collaborative performances, with musicians such as John Butcher (saxophone) and Gino Robair (percussion, electronics).

Video excerpts of the *Interstices* particle-based pieces can be found at

http://userwww.sfsu.edu/~whsu/PSHIVA/

Excerpts of the *Flayed/Flock* pieces can be found at http://userwww.sfsu.edu/~whsu/ARFlock/

5. REFERENCES

[1] Barri, T. Versum: audiovisual composing in 3d. In *Proceedings of the 15th International Conference on Auditory Display (ICAD2009)* (Copenhagen, Denmark, May 18-21, 2009).

[2] Bisig, B. and Neukom, M. Swarm Based Computer Music – Towards a Repertory of Strategies. In *Proceedings of the 11th Generative Art Conference* (GA 2008).

[3] Blackwell, T. Swarming and Music. In *Evolutionary Computer Music*, 2007, pp. 194 – 217, Springer-Verlag.

[4] Jehan, T., and Schoner, B. An Audio-Driven Perceptually Meaningful Timbre Synthesizer. In *Proceedings of the International Computer Music Conference* (2001).

[5] Malt, M. and Jourdan, E. Real-Time Uses of Low Level Sound Descriptors as Event Detection Functions Using the Max/MSP Zsa Descriptors Library. In *Proceedings of the 12th Brazilian Smposium on Computer Music* (September 2009).

[6] McDonald, K. Clouds are Looming, http://openprocessing.org/visuals/?visualID=6753

[7] Murphy, G. Smoke 2, http://bodytag.org/smoke2/

[8] Murphy, G. Fluid Bodies, http://bodytag.org/fluid_bodies/

[9] Rowe, R. and Singer, E. Two Highly Integrated Real-Time Music and Graphics Performance Systems. In *Proceedings of the International Computer Music Conference* (1997).

Creating Interactive Multimedia Works with Bio-data

Claudia Robles Angel
Freelance Media Artist
Dürenerstrasse 176 – 50931 – Cologne - Germany
+49 221 27783325. www.claudearobles.de
post@claudearobles.de

ABSTRACT
This paper deals with the usage of bio-data from performers to create interactive multimedia performances or installations. It presents this type of research in some art works produced in the last fifty years (such as Lucier's *Music for a Solo Performance,* from 1965), including two interactive performances of my authorship, which use two different types of bio-interfaces: on the one hand, an EMG (Electromyography) and on the other hand, an EEG (electroencephalography). The paper explores the interaction between the human body and real-time media (audio and visual) by the usage of bio-interfaces. This research is based on biofeedback investigations pursued by the psychologist Neal E. Miller in the 1960s, mainly based on finding new methods to reduce stress. However, this article explains and shows examples in which biofeedback research is used for artistic purposes only.

Keywords
Live electronics, Butoh, performance, biofeedback, interactive sound and video.

1. INTRODUCTION
The biofeedback method developed in the 1960s by the psychologist Neal E. Miller consists of the process of measuring physiological parameters from a subject (for example, the heartbeat, the brainwaves or the breathing), sending this data to a computer to be analysed and afterwards translating these parameters to sound and video, feeding them back to the subject and increasing body and mind awareness. Diebner explains that "[F]or example, we are consciously and only partially aware of the heart rate under physical strain or under cardiac arrhythmia. If, however, the heart rate is recorded and the signal transformed into sound or visuals, then this physiological process is accessible to our senses". [2] This treatment technique was originally created to improve health, helping tense and anxious people to learn how to alter these functions by relaxing.

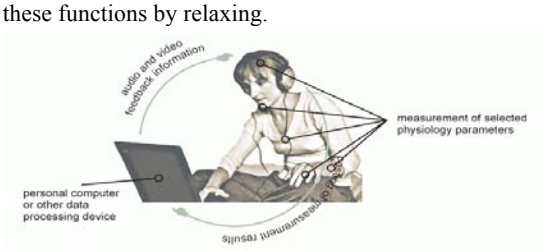


Figure 1. Visual representation of biofeedback
(http://en.wikipedia.org/wiki/Biofeedback)

There is a wide range of sensors used for the biofeedback methods such as the GSR (Galvanic Skin Response), the EMG (Electromyography), the EEG (electroencephalography), the ECG (Electrocardiography) and some others. The two bio-interfaces introduced in this paper are the EMG[1] and the EEG[2].

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. First Examples of the Usage of Bio-data
One of the first art works using bio-data is the famous brain wave piece by Alvin Lucier: *Music for a Solo Performance (1965).* For this piece, he attached some electrodes to the performer's scalp, measuring his brain activity (the alpha rhythm range from 8 to 12 Hz.) and sending these electrical signals to amplifiers and loudspeakers connected to a large set of percussions instruments. At that time, bio-feedback devices were not as sophisticated as nowadays, so he used EEG equipment belonging to the US Air Force and with the technical support of Edmon Dewan. "He generously lent me his apparatus, consisting of a pair of electrodes, a differential amplifier, and a band pass filter, set to a band-width just wide enough to let the ten Hertz alpha waves flow through and at the same time reject unwanted electrical and ambient noise." [4]

Figure 2. Music for a Solo Performer
(http://www.kunstradio.at/ZEITGLEICH/CATALOG/ENGLISH/lucier-e.html)

Since the mid 1960s, many composers and video artist such as, for example, David Rosenboom, Atau Tanaka, Yoichi Nagashima and Mariko Mori have experimented with bio-data to produce music, installations and interactive performances. From the EEG Air Force's equipment used by Alvin Lucier to the sensor-based musical instruments research by A. Tanaka in the last 20 years, the use of bio-signals to interact with the media (sound or/and video) continues to raise the interest of several artists worldwide, mostly seeing the accessibility, for example, via internet, of

[1] *The EMG signal is a biomedical signal that measures electrical currents generated in muscles during its contraction representing neuromuscular activities.* [5]

[2] *The electroencephalogram (EEG) is defined as electrical activity of an alternating type recorded from the scalp surface after being picked up by metal electrodes and conductive media.* [7]

several devices/software which are able to perform the data exchange (such as, for example, the *Arduino* project). The following two artworks introduced in this paper, ***Seed/Tree*** (an interactive Butoh performance-installation) and ***INsideOUT*** (an interactive multimedia performance) were produced using new accessible technologies and one of them (the EEG) with an open source hardware/software, was assembled without the need of expensive and difficult-to-access hardware by myself, and then programmed in real time with the software MAX/MSP-Jitter.

3. *SEED /TREE* (2005) Installation/Butoh Performance/Live Electronics

This project was created during an "artist in residence" program at the ZKM (Centre for Media Art in Karlsruhe, Germany).

Butoh is a modern expressive dance-form created in Japan in the 1960s by Tatsumi Hijikata and Kazuo Ohno. It is traditionally performed in white-body make-up and the movements are very slow and expressive. The movements in this dance form come from the inner world, they must emerge from within and not be imposed from without; Butoh is not a representative dance. During a performance, the Butoh dancer is in a state of 'hyper-presence', he is aware of everything around him and within his own body.

This installation-performance consists on a forest environment created by some panels projected by haptic images from tree cortex and human skin. By haptic images I mean: HAPTIC from (the Greek word: HAPTOS - tactile): the sensation "to feel": the feeling that one can 'touch' with the eyes, according to the French philosopher Gilles Deleuze: '*Where there is close vision, space is not visual, or rather the eye itself has a haptic, non optical function: no line separates earth from sky, which are of the same substance; there is neither horizon nor background nor perspective nor limit nor outline or form nor center; there is no intermediary distance, or all distance is intermediary.'* [1]

There are three Butoh dancers in the space performing the process of a seed growing to become a tree. Each performer moves in his/her own way representing the same subject a-synchronically. In the performance, the inner impulse of every dancer emerges on the stage; the dancers have each their own movements, which are the product of their own imagination.

The installation runs for three hours. The dancers develop the main subject in twenty minutes, then they lie for ten minutes on the floor and afterwards they repeat the process from the beginning. During this time, the performers have the necessary time to experience their own imaginary world combined with the outer space created by the sound and video projection. There is a continuous feedback between the dancers and the media; the translation of emotional physiological parameters to sound and video, however, gives feedback not only to the dancers but also to the spectators.

Feelings, associations, mental images and spontaneous impulses are the starting point for the creation of stories and choreography. In *Seed/Tree* the dancers produce and transform the sound and the audience control the video projections. The results are instantaneous creations, expressions of the moment, with image, movement and music forming living signs in space.

There are two types of interactivity in this performance. The first one is the interaction between dance and sound: the performers have microphones and EMG electrodes attached to their bodies. The breathing and the heartbeat of two of the performers produce sounds that are continuously modified by the muscular tension of a third dancer. The second type of interactivity is that between the installation space and the visitors. During the performance, visitors can walk freely around the virtual forest. There is a video observing the installation space and human presence influences the video projections; these interactions create subtle differences of the video on the panels. The installation space itself becomes aware of the visitors and reacts according to their movements. In this way the visitor is invited to be part of the environment.

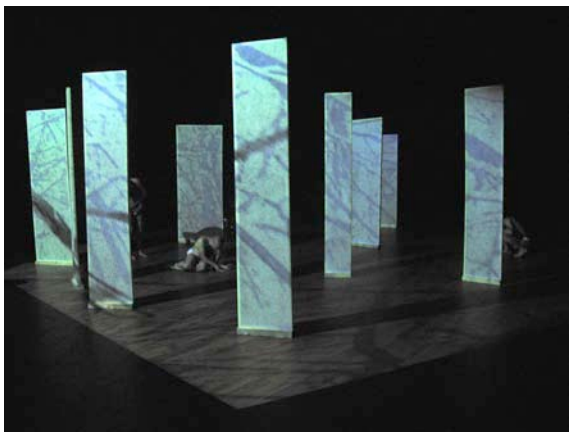


Figure 3. Seed/Tree at the ZKM Centre Karlsruhe (Germany)©, 2004

For *Seed/Tree* I used a wireless EMG (electromyogram) interface developed by Frieder Weiss, as shown in figure 4.

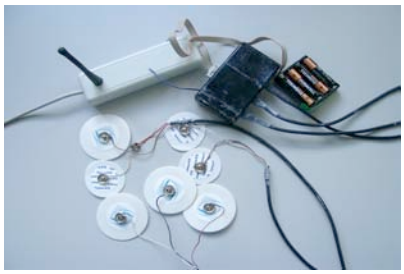


Figure 4. EMG Interface by Frieder Weiss

This interface has three pairs of electrodes, which are attached to three different muscles. This interface measures the muscle tension, while the program sends the values as a continuous OSC packet that is received through an OSC-route object in the MAX program; then, each value is used to trigger different sound effects in MAX/MSP.

Figure 5. Seed/Tree: EMG electrode attached to the Butoh dancer.

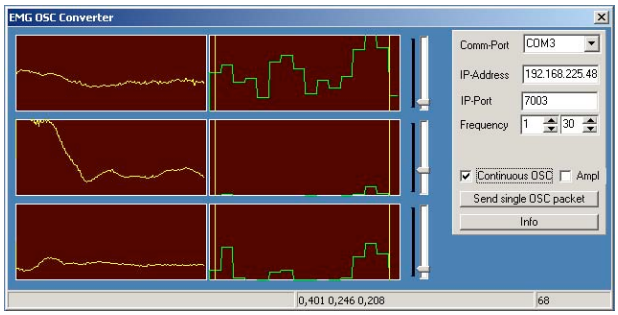


Figure 6. The Frieder Weiss program sending the values as a continuous OSC packet

Figure 7. Seed/Tree: the MAX/MSP patch by Claudia Robles.

4. *INsideOUT* (2009) Performance/Live Electronics

This project was created during an "artist in residence" program at the KHM (Academy of Media Arts in Cologne, Germany). It is about the materialization of the performer's thoughts and feelings on the stage. '*The stage is a place for the appearance of the invisible. Yasu Ohashi says: the actors aim at our senses, our body and our unconscious and not at our intellect. Their gestures try to envision THE INVISIBLE WORLD'* [3].

The performer interacts with the sound and images using an EEG (electroencephalogram) interface, which measures the performer's brain activity. Those sounds and images – some already stored in the computer and some produced live- are continuously modified by the values from two electrode combinations via MAX/MSP-Jitter. Hence, the performer determines how those combinations will be revealed to the audience. Images are projected to a screen and also onto the performer, while sounds are projected in surround.

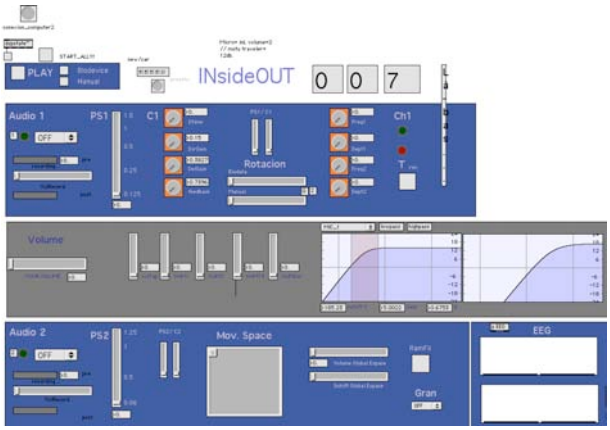


Figure 8. *INsideOUT:* MAX/MSP-Jitter patches by Claudia Robles

The German psychiatric doctor Hans Berger was the first to produce electroencephalograms of human subjects and also the first to observe the characteristic alpha-rhythm. The EEG project was meant at the beginning as the expression of the self, about turning the subject's imagination from the inside to the outside. For this performance, I used an open source EEG interface from Olimex, which measures the brain activity and consists of two assembled boards: one analogue and the other digital.

Figure 9. The analogue and digital boards from Olimex.[3]

Each board has two electrode combinations (or two EEG channels). It is possible to connect up to three boards (i.e. six EEG channels). I use only two for this piece: the frontal and the occipital channels. The rubber cap and the contact electrodes of the interface are those typically used in medical applications.

Figure 10. *INsideOUT* at the University of Miami's CAS Art Gallery (USA). The image shows how the electrodes are connected to the scalp. © 2009[4]

[3] Figure from: http://www.olimex.com/gadgets/index.html

[4] Photo by Javier A. Garavaglia

To adapt this interface I received technical support by Lasse Scherffig and Martin Nawrath at Lab3-KHM. They modified the open EEG device by replacing the Atmel microcontroller with one running the Arduino firmware and changing the quartz clock accordingly to 16 MHz.

Lasse Scherffig wrote a program with the software *Processing*, which reads the values of both channels from the open EEG via a serial communication. The modified open EEG sends ASCII-formatted data representing the voltages of both channels (at a frequency of 100 Hz). In Processing, a Fast Fourier Transform (FFT) is applied to that data and extracts the bins for the frequencies 0-50 Hz. From these, the median of the frequencies for the alpha channel (8-13 Hz) is extracted, smoothed using a low-pass filter and transmitted via OSC, which is received once again by the OSC-route object in the MAX program. This frequency band between 8-13 Hz correspond to the Alpha frequency range that is accentuated during relaxation.

Figure 11. *INsideOUT* at the SIGGRAPHAsia2009 Yokohama (Japan) © 2009[5]

For the performance of *INsideOUT* I have tried to train my brain in order to control the media combinations on the stage, putting in evidence different emotional and mental states, which cannot be achieved without the input of data coming from my own brain waves via the EEG interface. However, this conscious control is not completely attained due to the enormous and uncontrollable stream of feelings that generally appear surprisingly under such circumstances.

Thanks to new accessible technologies, the possibility to built complex interfaces, wireless and lighter as those used in medicine or research context, new forms of performances on the stage or new forms of relations between machine, human body and space have emerged. Lucier's performances in the 1960s and 1970s or David Rosenboom's brain analysis software (1976-77) used "*for creating self-organizing musical forms*" [6] are probably the first attempts to use bio-data for purposes other than scientific, in which they created new relationships between performers, the performers' brain activity, instruments and performing space, giving a new usage of EEG, creating as Rosenboom said: "a self-organizing, dynamical system, rather than a fixed musical composition". [6] In the case of Mariko Mori's *Wave UFO*, (1999–2002) there is a search for the representation of the Buddhist concept of *oneness*, by the interconnection between all three participants with each other, bringing them into a deeper state of consciousness, interconnecting the self and the universe. As it can be observed, there has been rather different approaches in the last fifty years in how this technology can be manipulated for artistic purposes.

This research and its artistic results aim to raise awareness of the human body and its functions (e.g. muscle tension, breathing, etc.) as a means to manipulating the media by controlling those functions consciously on the stage via the usage of bio-interfaces. This creates an environment that invites the visitor to perceive the body in a different way and to reflect on the relationship between the human and the machine. Following Ohashi's concept of the *invisible world* already mentioned before, this research and, most specifically, its artistic results should allow for an empty space that could be populated by the invisible or the imperceptible.

5. ACKNOWLEDGMENTS

I would like to thank Lasse Scherffig and Martin Nawrat at KHM (http://interface.khm.de) for their help in specific technical requirements for *INsideOUT*.

6. REFERENCES

[1] Deleuze, G., Guattari, F. and Massumi, B. 2004. *A thousand plateaus: capitalism and schizophrenia.* Continuum International Publishing Group, NYC, p. 545.

[2] Diebner, H. Taken from his home page accessed on 20.01.2011: http://diebner.de/htmldocs/biofeedback_en.html

[3] Haerdter, M. and Kawai, S. 1998. *Rebellion des Körpers, Butoh, ein Tanz aus Japan*. Alexander Verlag, Berlin, p. 25

[4] Lucier, A., Gronemeyer, G. and Oehlschlaegel, R. 1995. *REFLECTIONS Interview, Scores, Writings 1965–1994.* Edition MusikTexte, Koeln, p. 442.

[5] Reaz, M. B. I., Hussain, M. S. and Mohd-Yasin, F. 2006. *Techniques of EMG Signal Analysis: Detection, Processing, Classification and Applications.* Biological Procedures Online, vol. 8, issue 1, pp. 11–35.

[6] Rosenboom, D. 2000. liner notes, to *Invisible Gold*. PogusProductions.

[7] Teplan, M. 2002. *Fundamentals of EEG measurement*, Measurement Science Review, Volume 2, Section 2. DOI= http://www.measurement.sk/2002/S2/Teplan.pdf

[5] Photo by M. Goldowski

TresnaNet
Musical Generation based on Network Protocols

Paula Ustarroz
iMiLab
Plaça Canonge Rodó 5 Bj.2
Barcelona (Spain)
paula.ustarroz@gmail.com

ABSTRACT
TresnaNet explores the potential of Telematics as a generator of musical expressions. I pretend *to sound* the silent flow of information from the network.

This is realized through the fabrication of a prototype following the intention of giving substance to the intangible parameters of our communication. The result may have educational, commercial and artistic applications because it is a physical and perceptible representation of the transfer of information over the network. This paper describes the design, implementation and conclusions about TresnaNet.

Keywords
Interface, musical generation, telematics, network, musical instrument, network sniffer.

1. INTRODUCTION
During the 20th century the relationship between art, science and technology has been converging. From *intonarrumori* [5] to the *Brain Opera*, the different scientific and technological advances have changed the art and art has reflected it on their applications and also has found new uses.

The concepts of author, media and audience have changed in this rapid process, from the spectator and the static art object, to the interactor and the interactive art system. These technical and artistic changes can be understood from a new aesthetic perspective that Claudia Giannetti named endoaesthetics [3].

If art, science and technology have always been related, in some of the contemporary artistic productions they converge, dissolving and diluting our previous model of aesthetic experience.

1.1 New environment, rethinking art
Ars Telematica [2] defines in the same concept, all the art created using Internet as a tool. The salient features of this environment (understanding the Internet as a environment) are: connectivity (being online), non-hierarchical and interactive communication, global scale, immediacy (real time), human-machine interaction and body absence (cyberspace, telepresence). These characteristics of the environment, cause that Telematic Art perceives the art as **communication**. Highlighting the importance of communication over the content. We also have to consider two fundamental aspects of the art, **real-time** development and **global scale**.

The nature of their non-hierarchical and horizontal communications model made that Roy Ascott linked the connectionism paradigm with Telematic art. Thus, Internet can be understood as a neural network where the nodes reinforce the connections according to their interests, through their interaction and communication. This network is composed of multiple links and **connections**, such as neurons and connectors form the brain, producing in both cases an asymmetric and specialized connection. [1].

Which brings us to the definition of an art that differs from traditional concepts of author, audience and artwork to focus on interactive models and intelligent systems. Just as the network itself is a system, which provides an open and equal communication between their different nodes, the interactive art provides this same communication between transmitter-receiver-environment.

This is the goal of *TresnaNet*, a sound device that reflects the sound (or silence) of our time (Internet), as *intonarrumori* reflected the sound of the early 20th century, the machines.

2. KNOWING *TresnaNet*
TresnaNet is located as shown in Figure 1 and must be able to connect to the network via Ethernet cable or wireless (if the network offers that option). In this type of network connection, *TresnaNet* should act as a sniffer and monitor all network traffic. This can be a limitation because of the topology of the network.

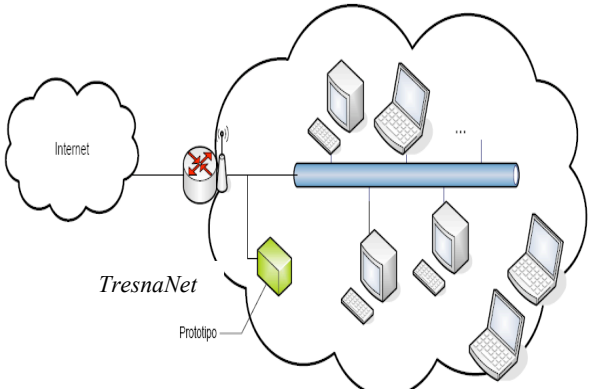


Figure 1. Network diagram that displays the location of TresnaNet.

As shown in the Figure 2, there is an inflow to the system from an external source (network) and an outflow (sound result) that depends on network data. It should consider the possible limitation caused by the type of data from the source, i.e., all flow into *TresnaNet* proceeds from a real time environment, and this stream of constantly changing data will have an impact on the output of the system

2.1 TresnaNet architecture

The desired architecture for the prototype is modular, dividing into layers the different functionalities of the prototype. Various processes are developed within each layer.

Figure 2 shows the intermediate steps between the network (source) and music (output): the physical layer (L1 NTS[1]) connects to the network and performs the extraction of raw materials; it deals with data at the lowest level and informs the user/performer. Secondly, the middle layer (L2 DMC[2]) performs a musical composition or dynamic mapping of data from the previous layer to an upper layer (L3 SG[3]) where musical composition is generated, by using templates of instruments, establishing communication with the user to initiate this process and finally obtain the sound result.

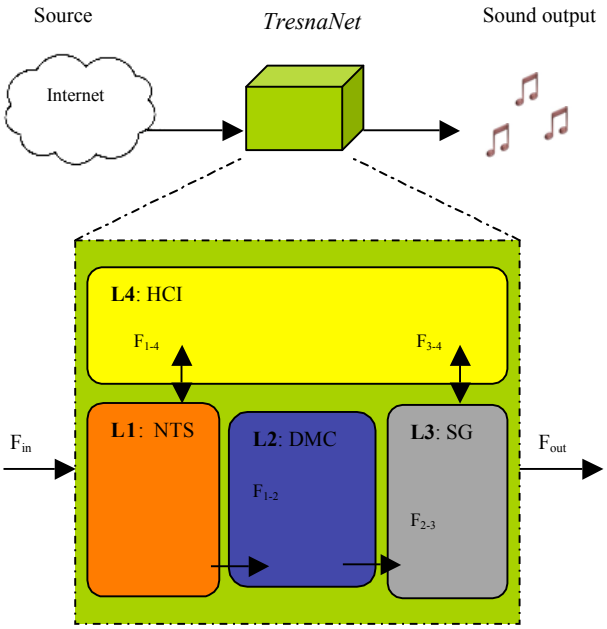


Figure 2. TresnaNet architecture diagram.

In parallel and interrelated with layers 1 and 3, is placed layer 4 (HCI[4]), which give the performance needed by the user for decision-making. The arrows in Figure 2 symbolize the different exchanges of data flows that take place in the system.

3. DESIGN

3.1 Networks and musical process

To access the information that travels and exists in a network must use a network sniffer; each network has similar characteristics and fundamental differences depending on their design, topography, servers, use, etc. To achieve an appropriate model for *TresnaNet*, we need to delimit the wide range of possibilities of using this information (traffic, raw material) and be able to understand this raw material extracting what interests us.

3.1.1 *Raw material*

A LAN[5] can be wireless or wired; therefore, to avoid limitations, *TresnaNet* extracts data from the network layer up. Among these data, can be highlighted: different protocols (network and transport layers), the routing of the packets (port mapping) and IP addresses.

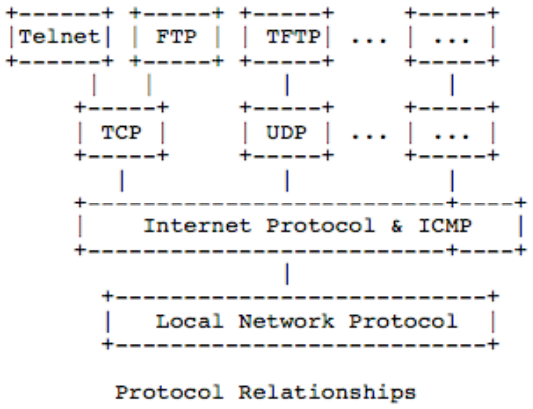


Figure 3. Protocol relationships from RFC 791.

3.1.2 *Sound typology*

What defines an Internet connection? What characterizes the traffic? How can we make equivalences or find a pattern?

Network traffic can be translated musically as a sound traffic noise with different characteristics, which refers to Schaeffer's idea of musical object [6]. But like everything, depends on the perspective of observation. If we want to define this traffic more generally, the object itself is equivalent to traffic, but if we want to highlight the features as a detail, each of them could be a musical object.

The reality is that network traffic is not made up of absolutely different objects, in other words, they have dependencies between them and what characterizes the traffic, for example in packet level, can be encapsulated into another one (TCP/IP levels).

Therefore, we can talk about the Matryoshka model or Russian doll, but repeatedly, as analogy. Thus, would have a set of objects with certain intersections or other typical operations of set theory. It is directly linked to the object-oriented programming and its distribution in classes and the methods of each of them.

Extrapolating back to a musical parallelism, data from network traffic provide a series of sound patterns (sets of objects); those can be understood as musical instruments (samplers) and may even resemble the mechanism of a synthesizer [4].

From there, TresnaNet create libraries of objects. In addition to the collections, must take into account certain global variables (featuring the network) that affect the characterization of the sound.

Some equivalences sound - telematics have been made during the development of TresnaNet, they are described in Table1.

[1] Network Traffic Sieve

[2] Dynamic Musical Composition

[3] Sound Generation

[4] Human Computer Interaction

[5] Local Area Network

Table 1. Sound – telematics equivalences

Sound features	Telematic equivalence
Reverb	Each time a network packet is created, a counter increases the number of packages and its number, influences the sound reverb.
Sound pan	The distribution of sound in space (L / R) depends on whether they are received or sent packets (i.e., if IP's that are maintaining a connection are repeated, these sent or received packets are distributed between left or right speaker).
Oscillators	The frequency is oscillated according to the resonance base. Used ports or the packet-length average defines the harmonic basis.
Rhythm	Based on the number of TCP/UDP packets type.
Specific sounds	Some specific services at the application level trigger certain sounds to make them recognizable. For example: - Trigger some sound effects on templates if port 80 is used. - Differentiate some web consulting, according to its IP: Facebook, Youtube, Hotmail, etc.
Sound repetitions (notes, sounds)	According to the IPs that sniffer detects.

3.2 Layer design

3.2.1 L1 NTS: Sniffer

Usually, this software is provided with a graphic interface for an easy view of data (packets) and decodes them. In this case, TresnaNet does not need a graphical interface, but the real-time capture and extraction is essential. Carnivore[6] performs this function. Processing fully integrated (it is a library) and exists the option of develop a specific *TresnaNet* sniffer, mapping data from the network into variables which are later sent to PD. Also because:

- It is Open Source.
- Captures, filters and transmits raw packets.
- Provides statistics.

[6] http://r-s-g.org/carnivore/ Developed by RSG. [Accessed, April 2008]

3.2.2 L3 SG: Pure Data[7]

Musical synthesis, either analog or digital, starts from scratch in sound generation. *TresnaNet* uses PD for layer 3, because of the following features:

- Open source and multiplatform.
- Communication with other software through OSC[8]
- Patches modification in real time.

3.2.3 L2 DMC and L4 HCI

TresnaNet uses Processing[9] as programming platform because it allows:

- Transmission by OSC with PD.
- To capture and to monitor sniffer data and customize it.
- To make a GUI with great interaction.
- A future communication with Arduino.

4. IMPLEMENTATION

TresnaNet made the extraction of network data through Carnivore; this network data reading is received by Processing. The program implemented, makes the mapping of these data, to pass through OSC to Pure Data, where sound generation happens. Using certain functions, the performer/user can interact with the artwork /*TresnaNet* through Arduino.

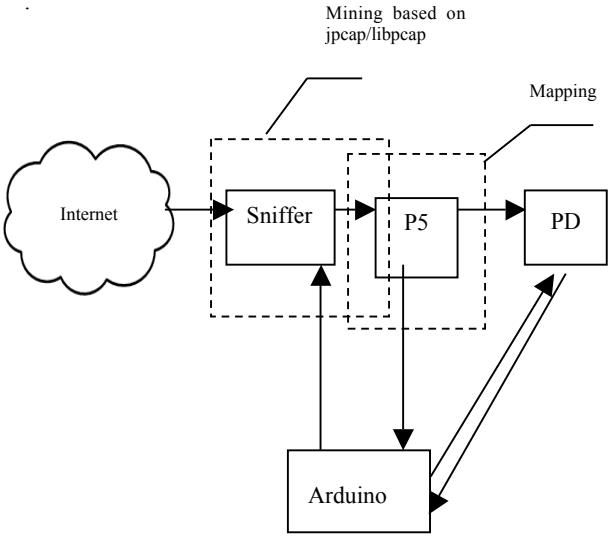


Figure 3. TresnaNet design diagram.

4.1 Processing

A modular logic is used to develop the code, in other words, different sketches for each feature:

- Sniffer: A customized Carnivore client has been developed to extract the desired data from the network packets and send them to PD.
- OSC communication: L2 sends messages to L3. IP must be defined and also the transmission received/sent port. In each message, a tag and a value must be assigned for PD interpretation.

[7] Henceforth PD. http:// puredata.info/ [Accessed, April 2008]

[8] Open Sound Control, http://opensoundcontrol.org/ [Accessed, April 2008]

[9] http://www.processing.org Processing [Accessed, April 2008]

- GUI[10]: designed to be as intuitive as possible from the start, made up of various buttons functionalities (sending some parameters of the network, trigger PureData, etc.) scrollbars for volume and ability to display network data. The classes are for buttons and scrollbar creation. And `method helpers` for displaying text on screen.

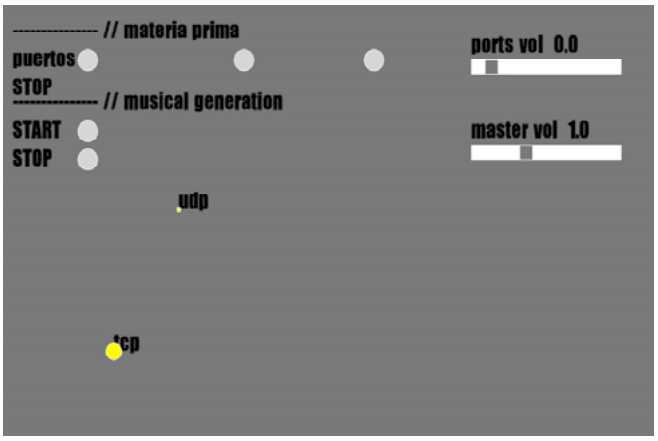


Figure 4. TresnaNet GUI screenshot.

4.2 PD Patches

An OSC client is implemented in the PD main patch in order to receive the data provided by Processing and to load the libraries that allow communication with OSC.

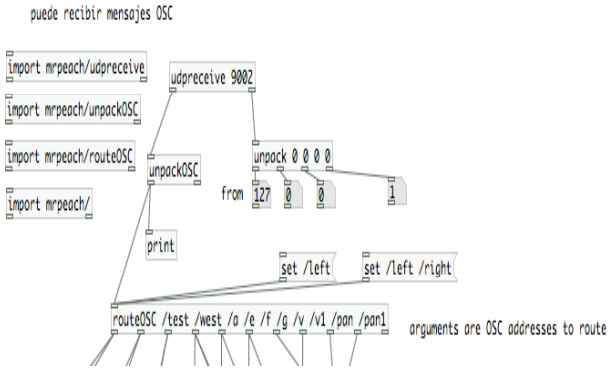


Figure 5. PD patch screenshot.

Also specify the UDP port where OSC packets are waiting, as shown in Figure 5. `RouteOSC` object provides the addressing of the arguments to our PD subpatches.

5. OUTCOME

5.1 Future research

HCI improvement. Currently all user interaction is through a graphical interface created with Processing. But the second phase of development of *TresnaNet* foresees the possibility of control by Wiimote. In addition, it is also planned to develop a controller using Arduino.

Evolve from a software gadget to an independent musical instrument. The above improvement is directly linked to it. If carried out, it should reconsider the *TresnaNet* architecture, considering the needs of processing (which are high) and interaction.

Thanks to this improvement, the user interaction (interpreter) or listener of the work would be more complete. The similarity to a traditional musical instrument would facilitate the understanding of the use and potential of *TresnaNet.*

Musical ensemble formed by different networks. Possibility of creating an installation would give the option of playing with several *TresnaNets* located on the same network or on remote networks.

Layer 1 utilization. As well as doing the extraction of the dedicated data to the musical composition, it also could route other data to visual generation. With this improvement, *TresnaNet* becomes into an audiovisual generation tool based on network protocols.

Research on Sound - Telematics equivalences. Continue to develop this model of equivalence and sustainable develop a theory of musical characterization.

5.2 Conclusions

The intersections between art, science and technology have been and are a reality, even now a necessity. The emerging art forms feed off the advanced technologies. In a society where being unconnected is beginning to look like science fiction, the understanding of the medium (Internet) is basic and telematics as a tool, can facilitate new forms of expression.

The way we communicate has changed in a few years, also the art, concepts as artwork, author, spectator, have mutated to pass to the interaction, leaving the mere spectator back. Similarly, the Web 2.0 is more than a reality; any design should be guided by these characteristics of change, interaction, and creation of own content, a great example would be the Reactable and in another level, this work.

TresnaNet can have an educational and artistic use, because it is capable of displaying or make sound the silence of existing networks, making perceptible the amount of data that travel around us and customize them.

6. REFERENCES

[1] Ascott, R. *Interactive Terminology. An Interfacial Glossary.* http://spark.com/who/ascotessay.html [Accessed, April 2008].

[2] Giannetti, C. *Ars telematica. Telecomunicación, Internet y Ciberespacio.* L'angelot, Barcelona, 1998

[3] Giannetti, C. *Estética digital: sintonía del arte, la ciencia y la tecnología.* A. C. C. L'angelot, Barcelona, 2002.

[4] Jordà, S. *Audio digital y MIDI.* GuíasMonográficas Anaya Multimedia, Madrid, 1997.

[5] Russolo L. *El arte de los ruidos.* Centro de Creación Experimental de la Universidad de Castilla-La Mancha. Cuenca, Spain, 1916.

[6] Schaeffer, P. *Tratado de los objetos musicales*. Alianza Editorial, Madrid, 1966.

7. APPENDIX

Soon all the documentation and music generated by *TresnaNet* in the iMiLab website[11].

[10] Graphical User Interface

[11] *impossibleMusicalinstrumentsLab* [underconstruction] http://impossibleinstruments.com

Designing a Music Performance Space for Persons with Intellectual Learning Disabilities

Matti Luhtala
VTT Technical Research Centre of Finland
P.O. BOX 1300,
33101 Tampere, Finland
matti.luhtala@vtt.fi

Tiina Kymäläinen
VTT Technical Research Centre of Finland
P.O. BOX 1300,
33101 Tampere, Finland
tiina.kymalainen@vtt.fi

Johan Plomp
VTT Technical Research Centre of Finland
P.O. BOX 1300,
33101 Tampere, Finland
johan.plomp@vtt.fi

ABSTRACT
This paper outlines the design and development process of the 'DIYSE Music Creation Tool' concept, by presenting key questions, the used methodology, the music instrument prototype development process and user research activities. The aim of this research is to study how music therapists (or instructors) can utilize novel technologies and study new performing opportunities in the music therapy context, with people who have intellectual learning disabilities.

The research applies an action research approach to develop new music technologies by co-designing with the music therapists, in order to develop in situ and improve the adoption of novel technologies. The proof-of-concept software utilizes Guitar Hero guitar controllers, and the software allows the music therapist to personalize interaction mappings between the physical and digital instrument components. By means of the guitars, the users are able to participate in various musical activities; they are able to play prepared musical compositions without extensive training, play together and perform for others. User research studies included the evaluation of the tool and research for performance opportunities.

Keywords
Music interfaces, music therapy, modifiable interfaces, design tools, Human-Technology Interaction (HTI), User-Centred Design (UCD), design for all (DfA), prototyping, performance.

1. INTRODUCTION
The Do It Yourself Smart Experiences (DIYSE) project[1] aims at enabling ordinary people to easily create setup and control applications in their smart living environments as well as in the public Internet-of-Things space, allowing them to leverage aware services and smart objects for obtaining highly personalized, social, interactive, flowing experiences at home and in the city. The development of the 'DIYSE Music Creation Tool' and the user research studies were based on a preliminary study within the DIYSE-project. The study outlined the everyday life of people with intellectual learning disabilities[2] concerning new technologies. Based on this preliminary research, the music therapy context was chosen for the research framework.

Learning to play a traditional musical instrument requires long-term training, and consequently many beginners never succeed to develop the necessary fine-motor skills to play music. This is especially the case with the end-user group of this study; people with intellectual learning disabilities. This user group needs musical interfaces that are extremely easy to understand and to adopt: the paradigm is also found in the studies of Machover [6] and Benvenieste [1]. Therefore, the aim of this study has been to design easy-to-learn interactive music instruments and develop alternative methods for music creation. This paper presents the challenges of interaction design related to the music creation context, and describes the prototype development and the user-centred design research processes [5].

[1] http://www.dyse.org
[2] People who have a mild or moderate intellectual learning disability (Diagnosis ICD-10).

2. INTERACTION DESIGN FOR A MUSIC CONTEXT
In the field of interaction design research, there is a demand for new design tools that enable creativity by means of explorative interaction, as opposed to limited executive and mission-based interaction (e.g. [3], [5] & [6]). Petersen et al. have proposed that the aesthetical interaction perspective offers an alternative to traditional interaction ideals [6]. In aesthetic interaction, the user is seen as an improvisator and the interaction between the human and the technology is a situation of play. According to Petersen, aesthetic interaction is found in the concept of intrigue that is connected to experience, surprise and serendipity in the use of interactive systems (ibid p. 274). In the light of Petersen's theory, equal attention should be paid to the players' cognitive skills, emotional values and bodily capabilities in the design of creativity-supporting music tools. A music-playing learning situation should enable the player to imagine, create, play, share and reflect on musical actions [7]. The playing situation involves an interaction feedback loop between the participants, their instruments and produced sounds. In an ideal state, a playing situation should encourage players to improvise and express themselves through playing and experiencing immersion.

3. METHODS AND TOOLS
3.1 Prototyping
In the pursue of finding means to support the musical activities of persons with intellectual learning disabilities, we began by simplifying the music creation process and concentrated on finding interactive technologies that were easily available and easy to use. According to the preliminary research, we had learned that the target group end-users had various, and often multiple, disabilities and that they were enthusiastic about

music. In the initial research phase, various sensor technologies were tried out and observed with the end-users. The prototyping phase was carried out through an iterative co-design process between the designers and a music therapist. The co-operation with the professional music therapist was an essential part of developing the prototype. For developing the digital user interface we used the Max MSP graphical programming language [9]. Nintendo Wii Remotes [10] and Guitar Hero controllers were chosen for our physical controller framework. Both of the technologies offered good technological support for realizing proof of concept prototypes because of their reliability and active open source and sharing communities.

3.2 Music Therapy Context

The use of the 'DIYSE Music Creation Tool' was observed and evaluated in a natural music therapy context (see figure 1), in order to gather information about the adoption and usability of the software and the instruments. Rinnekoti Foundation, a service provider for disabled people and partner in the project, provided the facilities for the evaluation of the 'DIYSE Music Creation Tool': a computer, three guitars and the software, were brought to the music therapy studio. The therapist chose the players based on their capability to benefit from the new means to make music and based on their availability for the whole observation period. On proposal of the music therapist, the observation period culminated into a final concert, in which the participants performed the music piece for an audience with the instruments accompanied by an acoustic drum kit. The concert was part of the DIYSE project research consortium meeting (see video link in the appendices section).

Figure 1. 'Music therapy session: learning to play and practicing for a performance.

3.3 Prototype Evaluations

The 'DIYSE Music Creation Tool'was evaluated in two phases. The first evaluation session was arranged in August 2010, at the Rinnekoti Foundation, Espoo, Finland. The participants were 26 – 58 years of age. All of the interviewees knew each other beforehand and were accustomed to participate in music therapy sessions. The research methods included observations and semi-structured interviews [4] and there were two objectives for the evaluations. Firstly, the initial goal was to determine technical requirements by utilizing co-design, and therefore the music software was introduced to the music therapist. Secondly, the acceptance and the user experience were evaluated with the players. At the end of the evaluation session, there was a short 'Sonic Sketching' workshop that was aimed to encourage participants to innovate surprising and inspiring ideas for novel music instruments.

The second evaluation phase was held between October/November 2010, at the Rinnekoti Foundation and the participants were mostly the same as in the first phase. The observation framework was arranged for 1.5 weeks observation period, and the music therapist was responsible for the therapy context within the given framework. The therapist carried out most of the therapy sessions individually. The video observation period lasted ten days, and seven music therapy sessions were video-recorded in that time scale. The recorded video material was analyzed based on the 'interaction analysis lab' method [2]. In the method, the observers comment about the context of the video material, create a hypothesis about what is occurring in the recording, and discuss about the context [8]. During the analysis, the material was observed according to four topics: supporting creativity, learning, user frustration and independent playing.

4. RESULTS

4.1 Prototype

The software features the following three functionalities: 1. Composing and restoring music tracks in the software. 2. Design of interaction mapping strategies between the guitar's interface elements and played sounds. 3. Choose sounds for the guitar. Figure 2 presents the software's interface layout.

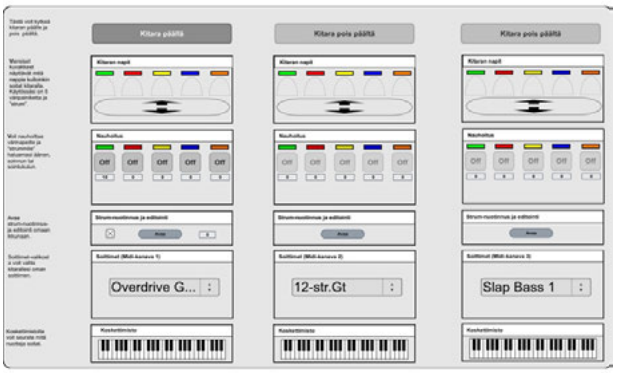


Figure 2. 'DIYSE Music Creation Tool' software's main window: functionalities for recording, mapping interactions and choosing sounds.

4.1.1 Prototype in Use

The music therapist prepared the therapy sessions by recording song arrangements into the software using a midi keyboard. The music in the first evaluation session was a general 12 bar 'blues' theme. For the second session, the therapist chose a song composed called 'Egyptian Reggae', by Richman Jonathan. The first song was used for practicing purposes and the latter was rehearsed to be performed in the concert. According to the therapist, the chosen genres – the blues and the reggae – were stated to provide a 'good groove' and were therefore suitable for the therapy group. In addition, the songs' musical elements were easy to arrange for the three guitars. The therapist arranged the musical elements as follows: 1. bass guitar, 2. rhythm guitar and 3. solo guitar.

The software assisted to map the arranged sound elements to the three guitars interface elements. The bass guitar was played

using the controller's strum switch. The switch triggered single notes from a step-sequencer's timeline in sequential order. The bass guitar was meant to be the easiest instrument to learn as it produced meaningful musical structures through simple switch triggering. The rhythm guitar and the solo guitar were played by pressing the color buttons attached on the controller's neck. The rhythm guitar's idea was to challenge the player to play chords in the right order and time. The solo guitar was designed to support an idea of free playing and expression. Using only harmonic notes, each mapped to the guitar's colored buttons, the audible result was designed to be pleasing as there were no dissonant notes.

4.2 Interviews and Observations

According to the interviews and observations, the most satisfactory attribute of the 'DIYSE Music Creation Tool' was the experience itself; the joy of creating and generating music and the feel of accomplishing something in a short period of time, even if the players lacked the skills to play musical instruments. This sense of easiness was consequence of the fact that some music pieces were composed beforehand and thereby there were "no wrong notes" i.e. if the player pressed the bass guitar's strum, the music flowed and sounded pleasantly. According to the preliminary observations, it seemed to be important that the instruments resembled real instruments, guitar and bass, so that its affordances were easy to perceive [2]. For the music therapist, it was important that there were many alternatives to choose from the sound library, relating to music genres, instruments, sounds and tones. The most significant finding was the fact that performing to an audience seemed to be important for this user group. Generally, the threshold to perform and try out new things seemed to be quite low.

4.3 Interaction Analysis Lab

The music therapist used much effort in trying to provide a creative atmosphere, so that the therapy situation would not be just about pressing buttons and learning rhythm. For example, he accompanied the players by playing traditional instruments and encouraged the players to communicate with him through musical expressions such as tempo variations and pauses. In addition, he made occasional polyrhythmic textures in order to increase the complexity level of playing. Many times his efforts disturbed the participants, as finding the rhythm took all their attention. Otherwise, the playing situation was quite static; it appeared that there was not much improvisation or experimental playing during the practise. An incentive to support creativity with the 'DIYSE Music Creation Tool' was the promised performance for an audience.

The observations indicated that the appearances, the shape and sound of the instrument, were important and that the instruments must support the player's identity. For example, one of the players mentioned that because his brother played the guitar in a band, he liked to play it too. However, the guitar-like shape also provided challenges: it was difficult to detect the colour buttons and it was challenging to decide how to hold the instrument, as it seemed to be uncomfortable to hold it 'like a guitar'. Some participants even did not have enough motor coordination to play the instrument like a guitar. This was an important observation, because the way to hold the instrument influences the way feedback is received. Preferably, the interaction with the instrument should be as intuitive as possible. Observations indicated also, that it seems to be more important for the players to press the right button at the right time, than to have a subjective playing experience and feel comfortable in the role of an improviser.

The music therapist himself learnt to prepare the system on the third observation day; connecting the guitars and the computer, uploading the sounds and creating personalized mapping strategies for the players. The most significant observed difficulty of the learning experience was related to learning the rhythm. If the players could not find the rhythm, it became difficult to perceive a mental map of the overall situation, and the users were disappointed and frustrated. In general, the players of this user group needed a lot of support from the therapist i.e. the level of independency was low. The music therapist guided the participants e.g. by instructing the colour keys of the instrument: "red-green-yellow" (see figure 3). On the third observation day, there was a new player attending the music therapy sessions. He practiced playing the instruments only once and was therefore an excellent subject to study. At first, he played the bass and was able to learn the first three notes of the rhythm pattern, but learning the whole rhythm structure seemed to be quite demanding for him too. Yet when he finally had learned the rhythm, it seemed to be extremely rewarding.

Figure 3. Practicing to operate the instruments: music therapist giving instructions – red, green, yellow...

During the observations, there were specific moments when players seemed to be quite frustrated. For example, in the second observation day one of the players was notably disturbed. His playing of the bass was already fluent and therefore his gaze wondered towards other interests. One of the players stated to be tired of the chosen piece of music, and he wanted to play something else. Frustration was apparent especially when the participant's ability to learn was not properly taken into account. There seemed to be a delicate balance between patronising the player and providing too much information and encouragement for independent playing.

5. DISCUSSION

The 'DIYSE Music Creation Tool' was intended to be a design tool for the music therapist. During the design process, the therapist utilized the system for planning the sessions for his customers, and the end-users utilized the system for playing music and performing. By co-designing the system and using it in a real therapy situation, it was possible to create and develop new music playing experiences for music therapy clients in situ. Creativity was chosen to be a critical issue of the study. Based on the theoretical background and the results attained through the user evaluations, it was perceived that the Guitar Hero controller is not an ideal controller for playing music. The controller's interface elements mainly support point and click

interaction style, which is suitable for playing rhythm games as indicated by Machover [5]. Supporting only the rhythm is not nearly enough; rhythm, timbre, pitch and time should all be considered equally important when designing interactive music instruments. In the light of Petersen et al [6], the Guitar Hero controller can be mainly seen from the mechanistic tool perspective, thus having distance to dialogue, media and aesthetic views of interaction. In search of the aesthetic experience, we emphasize experimental aspects of the four music elements presented above. Therefore interaction design of the instrument should encourage the player to explore and playfully appropriate the musical dimensions through the instrument. However, it must be acknowledged that the point and click interaction is one considerable alternative when designing music instruments for persons with learning disabilities. A significant finding of the research was that it is important to minimize the possibilities to fail (or the feeling of failure) by keeping the control of the instrument simple. On the other hand, it is important that the playing situation challenge the player in the five learning phases that Resnic [7] presents: imaging, creating, playing, sharing and reflecting.

In future research, we intend to develop instruments that enable explorative human-computer interaction. This allows the players to concentrate on the creative process of music making and creating in a performance space. Performing on stage and training for the performance were stated to be very important. Some of the challenges for future design phases include providing support for two or more players and for the co-playing concept as a whole. Social media could support in developing the music performance space by offering a tool for publishing music and providing a place for recording music or performing. In an ideal situation, digital and physical tools help users to enhance their everyday life; to think, to design and create art, experiment with new technology and technological gadgets and become stakeholders in public projects.

6. ACKNOWLEDGMENTS

This work was done as a part of the Eureka/ITEA2 DIYSE project in a cooperation between the Technological Research Center of Finland (VTT), the Rinnekoti Foundation and Laurea University of Applied Sciences. We gratefully acknowledge the financial support by the Ubicom programme of Tekes.

7. REFERENCES

[1] Benvenieste, J. Jouvelot, P., Lecourt, E. and Renaud, M. Designing Wiiprovisation for Mediation in Group Music Therapy with Children Suffering from Behavioral Disorders, IDC 2009, Como Italy.

[2] Jordan, B. & Henderson, A., 1995. Interaction analysis: Foundations and practitice, The journal of the learning sciences 4 (1), 39-103.

[3] Krippendorf, K. 2006. The semantic turn, a new Foundation for design, Taylor & Francis.

[4] Kuniavsky, M., 2003. Observing the user experience: a practitioner's guide to user research, Morgan Kaufmann.

[5] Machover, T. 2009. "Beyond Guitar Hero - Towards a New Musical Ecology." *RSA Journal* (London), January–March 2009.

[6] Petersen, M. Iversen, o. Krogh, P and Luvigse, M. Aesthetic Interaction – A pragmatist's Aesthetics of Interactive Systems. DIS2004, Cambridge, Massachusetts, USA.

[7] Resnic, M, N/A, All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten, Mitchel Resnick: kindergarten approach to learning, MIT Media Lab, Cambridge.

[8] Ylirisku, S., Buur, J., 2007. Designing with video: Focusing the User-centred design process. Springer.

Links:

[9] Max MSP, Web site (read April 26, 2011): http://www.cycling74.com

[10] Nintendo Wii Remote, Web site (read April 26, 2011): http://www.nintendo.com/wii/console/controllers

Video

http://www.youtube.com/HTIforWelbeing

Raja - A Multidisciplinary Artistic Performance

Tom Ahola
Nokia Research Center
Itämerenkau 11-13
00180 Helsinki, Finland
tom.m.ahola@nokia.com

Koray Tahiroğlu
Aalto University,
School of Art and Design,
Department of Media
PO box 31000 00076 Aalto, Finland
koray.tahiroglu@aalto.fi

Teemu Ahmaniemi
Nokia Research Center
Itämerenkau 11-13
00180 Helsinki, Finland
teemu.ahmaniemi@nokia.com

Fabio Belloni
Nokia Research Center
Otakaari 5, Espoo, Finland
fabio.belloni@nokia.com

Ville Ranki
Nokia Research Center
Otakaari 5, Espoo, Finland
ville.v.ranki@nokia.com

ABSTRACT

Motion-based interactive systems have long been utilized in contemporary dance performances. These performances bring new insight to sound-action experiences in multidisciplinary art forms. This paper discusses the related technology within the framework of the dance piece, *Raja*. The performance set up of Raja gives a possibility to use two complementary tracking systems and two alternative choices for motion sensors in real-time audio-visual synthesis.

Keywords

raja, performance, dance, motion sensor, accelerometer, gyro, positioning, sonification, pure data, visualization, Qt

1. INTRODUCTION

Raja is a Finnish word and means border. We wanted to cross the border and tear down walls between interdisciplinary teams by joining technology and different art forms together. Raja is a new fusion example of technology, dance, sound design and computer graphics. It has been so far performed three times, in Tampere, Helsinki and London.

Human motion has been used to control sound in many different ways before. Commonly used technologies for detecting motion, position, proximity and gestures include camera based systems[3], light sensors, ultrasonic sensors[6], capacitive proximity sensors, piezo triggers, gyros[1] and accelerometers. The scope can be anything from tracking one solo performer to tracking large audiences[4]. The level of motion tracking varies from bistable acceleration triggers to accurate gesture recognition. The novelty in the Raja performance compared to other systems is the simultaneous use of two different tracking systems: motion and position. Also, there is a possibility to use both accelerometer and gyro for motion tracking. In the following sections the overview of the dance, the technical system, motion tracking and positioning technologies are presented. Strategies for sound synthesis and visualization are described in detail. The paper concludes with a presentation of outcomes and discussion about the future development of Raja.

2. THE CHOREOGRAPHY

The choreography of the Raja dance performance for three dancers (Figure 1) was sketched by a professional choreographer in an iterative process. Initially, the choreographer was briefed about the possibilities of the technologies to be used. At this stage, the choreographer created the first version of the choreography, which was used as a basis for further development.

Figure 1: The Raja dancers.

In the second phase, the sensor configuration was selected: the dancers would each wear one motion tracker on their right wrist and one positioning tag in their hair. This selection had an impact into some details of the choreography. For example, the hand positions and movements were emphasized in order to provide a better signal for the sound synthesis. In addition, the movements of the dancers were modified so that their position on the dance floor would have a greater spread and variety. The first version of the sound design was also introduced.

The third version of the performance was modified from the second version based on feedback from the audience. The performance was divided into three parts where different sets of sounds were used to create an interesting contrast between the parts. The middle part was slow and peaceful with some improvisation. The performance ended with a short aggressive section where the dancers were dancing very intensively.

3. THE TECHNICAL SYSTEM

The technical configuration (Figure 2) of the system includes three laptop computers, networked together via an Ethernet switch. The indoor positioning, motion sensing and sound synthesis systems were each using their own laptop. In theory, everything could run in a single computer. However, to be able to monitor and control the different technologies at the same time, it was more convenient that each team had their own laptop.

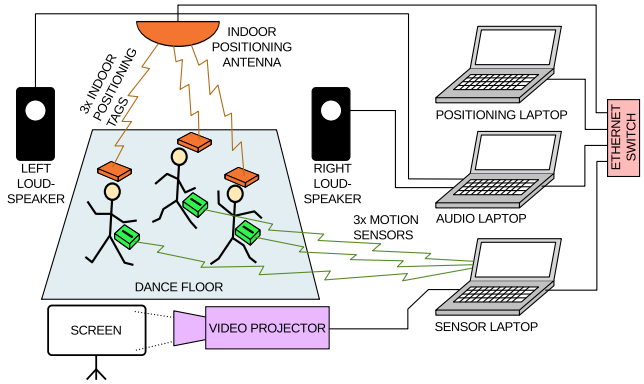


Figure 2: The performance system setup.

3.1 Indoor Positioning

High Accuracy Indoor Positioning (HAIP) technology developed in Nokia Research Center Helsinki Laboratory was used in the system[2]. HAIP provides position estimation of small battery powered transmitter tags by measuring the direction of the received UHF radio signals received by a receiver. The tags are worn by the dancers and the receiver is mounted above the dance floor. Several tags can be tracked with one receiver system. The accuracy of the positioning system is limited by the fact that the radio signal is blocked by the human body. Thus, the tags should preferably be mounted in the head area and the receiver should be mounted high up to have the best possible connection. In practice, the usable area is directly below the receiver so that the radius of the area is approximately 1.2 times the difference between the receiver and the tag height.

3.2 Motion Sensors

Ariane sensor-box motion sensors were used in the system. These are wireless sensor devices designed in the Advanced Systems Engineering department of Nokia Research Center. The sensor-box features accelerometer, gyro and magnetometer sensors, each with 3 axes. In addition they have two buttons, a light sensor, a barometer, a RGB LED, a vibro-tactile actuator and Bluetooth wireless connectivity. In Raja a $4g$ range is used for the accelerometer and the range of the gyro is 600°/s.

For the reception of sensor data from the motion sensors, an application called *SensorPerformer* was implemented. This application includes the sensor processing algorithms, a network interface to stream motion sensor data out, a network interface to receive position data, a visualization engine and a sound engine. The application was coded in C++ using the Qt multi-platform application development framework.

4. SENSOR SIGNAL PROCESSING

4.1 Accelerometer

4.1.1 *Effect of gravity*

The accelerometer was chosen as the motion sensor for the first version of the system. The challenge with an accelerometer for motion sensing is that it is also sensitive to the gravitational force. An algorithm was developed to remove the gravitation component from the sensor signal so that linear motion of the device could be used as the input to sound generation and visual presentation.

A device (Figure 3(a)) is affected by gravitational acceleration $\mathbf{a}_g$. Motion related linear acceleration $\mathbf{a}_m$ sums to the $\mathbf{a}_g$ to give a total acceleration $\mathbf{a}$, which is sensed by the accelerometer. We are only interested in motion and thus, if we know the gravitation we can remove it from the sensor signal and we get $\mathbf{a}_m = \mathbf{a} - \mathbf{a}_g$.

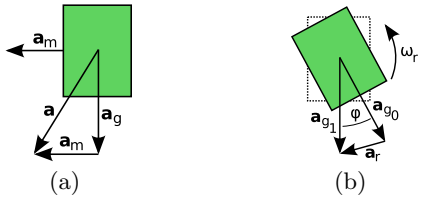


Figure 3: Total acceleration vector during linear movement (a) is the sum of gravitational acceleration and acceleration of the movement. Rotation (b) causes a change in the acceleration vector.

However, rotation of the device (Figure 3(b)) causes the gravity vector to change. In the initial orientation the gravitational acceleration is $\mathbf{a}_{g_0}$ and after rotation of angle φ the acceleration is $\mathbf{a}_{g_1}$. Thus, if we at some point assume gravity is $\mathbf{a}_{g_0}$ and remove it from the sensor signal to detect linear motion, after a rotation of φ the difference acceleration $\mathbf{a}_r = \mathbf{a}_{g_1} - \mathbf{a}_{g_0}$, is erroneously detected as motion. The magnitude of this error can be calculated as $|\mathbf{a}_r|^2 = 2g^2(1 - \cos\varphi)$, where $g = |\mathbf{a}_{g_0}| = |\mathbf{a}_{g_1}|$. A significant visible movement has an acceleration of $\Delta|\mathbf{a}_r| = 0.2g$, or more. An almost unnoticeable rotation of only $\Delta\varphi = \cos^{-1}(1 - \Delta|\mathbf{a}_r|^2/(2g^2)) \approx 11°$ results in such a change in the movement acceleration vector magnitude.

4.1.2 *Removal of gravity*

The red trace in Figure 4 shows the magnitude of the total acceleration vector measured from the wrist during a short segment of dance. In the beginning there are slow moves which build up to an energetic section. One can see that the acceleration approaches the $1g$ gravity in still parts.

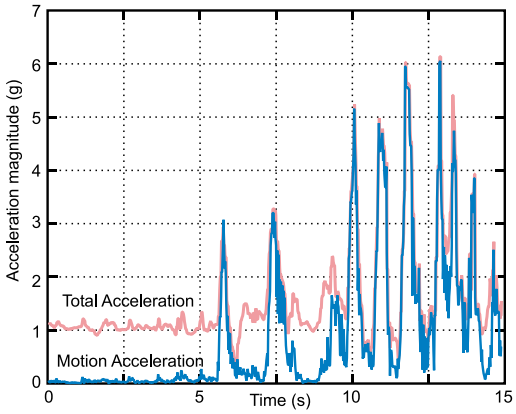


Figure 4: Accelerometer signal.

The removal of the gravity component is described in the diagram in Figure 5. Each orthogonal axis x, y and z of the acceleration vector is processed identically according to this algorithm. First the estimated gravity is subtracted from the signal so that the acceleration caused by movement remains. This signal is low-pass filtered to reduce noise before it is differentiated for detection of change. A comparator checks if the change is above a fixed threshold.

In moments of stillness the change signal is below the threshold and a switch enables integration of the movement acceleration. This integrator output is the estimate of the gravity component as the closed loop with negative feedback

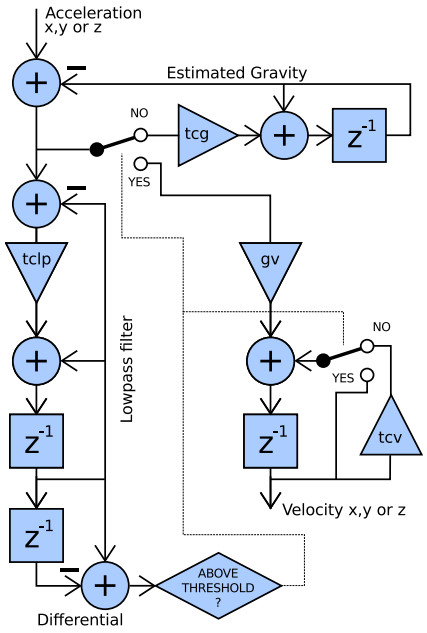


Figure 5: Gravity removal algorithm and velocity computation.

will settle so that the integrator input will be nulled. The blue trace in Figure 4 shows the motion acceleration, which is the result of gravity removal. It can be noticed that in most still moments the acceleration signal is now close to zero.

4.1.3 Computation of velocity

A velocity value is a more useful parameter because in most, if not all, natural sound generation processes the sound is related to the velocity of some mechanical part of the instrument, or air flow velocity in case of wind instruments. Acceleration happens mostly at the beginning of a move and at the end of a move, in the opposite direction. This would mean two separate signals for one sound gesture. By integrating the acceleration a velocity signal is generated with a smooth attack, sustain and decay.

During active movement the change signal (Figure 5) is above the threshold and a switch connects the motion acceleration signal to an integrator, which outputs a velocity value. In moments of stillness a switch leaks the integrator to reduce velocity offsets caused by asymmetry in the acceleration signal. Figure 6 shows the linear velocity computed from the acceleration shown in Figure 4.

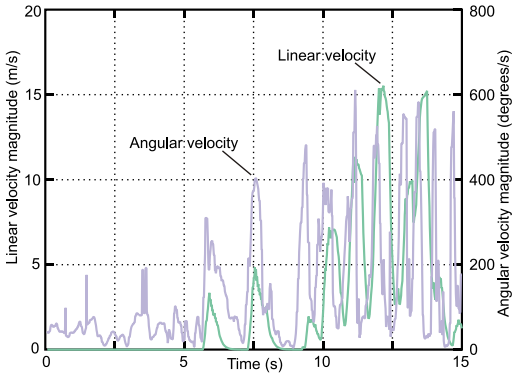


Figure 6: Velocity signal.

4.2 Gyro

We decided to use the gyro sensor instead of the accelerometer in the second performance set up as it is not influenced by gravity and the output is a velocity. We believed this would result in a more responsive and predictable sound representation of movement. The angular velocity trace in Figure 6 shows the gyro signal vector magnitude from the same wrist and performance as the accelerometer signals. Using the gyro signal was indeed found to be more responsive. A significant difference was observed, however, in the dynamic range. The gyro signal appears to be only little stronger during the energetic sections compared to the calm moments. We believe this is due to human physiology and motorics. Energetic movement results in only little increase in the rotational velocity of the wrist although linear acceleration can increase significantly.

5. AUDIO SYNTHESIS

The audio synthesis was implemented with Pure Data (PD). Sounds can be controlled by either angular or linear velocity, chosen independently between sensors. Three synthesized instruments were created, one for each dancer. The amplitude of each instrument is controlled by the vector magnitude of the velocity data so that the sound level correlates to intensity of movement.

The first instrument, *Frequency Synthesis Module*, maps musical textures with glassy, oscillating sounds. Eight digital oscillators' frequencies are controlled by the three velocity values from the 3-axis motion sensor. The 3-to-8 divergent mapping of the control signals is implemented so that each velocity controls each frequency with a certain weight. These weights were experimentally and artistically selected.

The second instrument, *Wave Module* is a polyphonic sampler with eight sample-voices. The magnitude of the velocity data applies parameter changes to the transformations of sampled sounds, controlling the playback rate of the polyphonic sampler. Resulted output implies dry, mechanical sounds with filtered pitch tonality.

The third instrument, *Sin Module*, is a frequency modulated oscillator. It generates a cosine wave with the amplitude controlled by an envelope generator. The magnitude of the velocity data is mapped to frequency and the envelop generator values. The output is streamed to a cosine waveshaper and filtered by a voltage controlled bandpass. The instrument generates brassy, sharp sounds with the dancer's movements.

In the third version of the Raja performance, a second set of instruments was introduced. The aim was to make this set towards more easy listening using classical instrument sounds. Piano samples were used for this set, which was used in the slower and more relaxed middle part of the performance, while keeping the electronic sound set in the beginning and end parts. The magnitude of the velocity data is scaled to integer numbers between 0 and 12. The generated values are mapped to MIDI note values in harmonic C minor scale.

Two separate background instruments were designed to support each section in the Raja performance. Both background instruments are sample based and create a dynamic rhythmic pattern. While the first background instrument creates certain tones continuously including the fundamental frequency in the piano section, the second one creates sounds containing inharmonic clusters of partials.

The positions of the dancers are mapped to the spatialization of each instrument. The system supports stereo or multichannel speaker systems. VBAP technology is used to control the direction of the audio stream in the perfor-

mance[5].

6. VISUALIZATION

An early version of the visualization was based on the idea that it would become a painting of the dance performance. Every movement and position of each of the dancers would be recorded in the painting and in theory the performance could be reconstructed from the painting afterwards. However, the visualization in this way was believed to be too chaotic for the audience to understand the correlation between dancers' actions and what was happening on the screen. To move closer to being a real-time illustration of position and movement the image was made to continuously fade away so that the more current graphical draw would stand out.

The final visualization design is implemented as follows. Three graphical objects of different color represent the three dancers. The positions of the objects on the screen are a direct mapping of the positions of the dancers on the dance floor. The objects can morph from a small dot to a large 8-bladed star when the dancers go from motionless to full motion. The star also rotates, with rotation direction and speed correlating with the sensed motion velocity. As a result each dancer is drawing a colored trace on the drawing canvas with size and texture matching the energy of the moves (Figure 7). And when the dancers stop, the traces fade away and only dots remain to indicate their position on the floor. The visualization was implemented using polygon drawing methods of Qt.

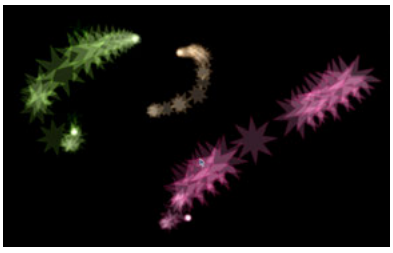


Figure 7: Visualization screenshot

7. THE EXPERIENCE OF THE DANCERS

We wanted to investigate how the dancers experienced the interactive sound production by asking them to fill in a questionnaire with three aspects: 1) implementation of the choreography when compared to a typical contemporary dance performance, 2) sensibility of the improvisation section and 3) the predictability of the sounds and implications of it on the dance.

From the dancers' perspective the main difference to a typical contemporary dance was the lack of the support from the music. Because it was missing, more communication between the dancers was required. Performing the movements simultaneously was more difficult because the synchronization had to be based only on visual communication between the dancers. On the other hand, missing time line of the music gave more temporal freedom in implementing the choreography, for example, the length of a pause.

The improvisation was considered to be the most pleasant part of the dance, especially because it used the classical piano sounds. After learning how the sounds respond to the movements, it was easier to modify the improvisation to produce certain type of sound patterns.

All the dancers mentioned that initially the sounds were somewhat irresponsive, predictability was weak and in some parts the sounds were not in harmony with the movement. These concerns, however, were dispelled when the project proceeded.

8. CONCLUSIONS

Comparing gyro and accelerometer sensors it was found in practice that the gyro sensor was more responsive and predictable. However, the dynamics of the gyro was flatter. The accelerometer gave a better match between the energy of movement and energy of the sound. Gravity removal and velocity computation from the acceleration signal is approximate and the cause of the drawbacks of the accelerometer. Human physiology, on the other hand, is the probable cause of reduced dynamics of using the gyro. We are currently working on an improved algorithm using both gyro and accelerometer to achieve a solution with benefits from both sensors.

The approach to produce sound by movement was interesting for the dancers. They especially liked the improvisation part and experienced it as an additional channel to express movements. When the dancers got freedom to plan the movements together with the sounds, the overall experience became more harmonious. Thus, the choreography and the sounds should be designed in parallel.

One of the future directions to improve Raja is to enhance the sound-action strategies. By recognizing different types of movements and mapping that to a sound synthesis engine with a richer set of timbral and temporal parameters a more intriguing experience for both performers and audience can be created.

9. ACKNOWLEDGMENTS

The authors would like to warmly thank the support from Tekes (HEI project) and the Academy of Finland (pr. 137646). Without this support this multidisciplinary collaboration work would not have been possible. We also thank choreographer Jari Saarelainen and dancers Kiia Elonen, Nora Laitinen, Mia-Mari Sinkkonen and Nina Kivisilta for their great creative co-operation.

10. REFERENCES

[1] R. Aylward and J. A. Paradiso. Sensemble: A wireless, compact, multi-user sensor system for interactive dance. *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06), Paris, France*, pages 134–139, 2006.

[2] F. Belloni, V. Ranki, A. Kainulainen, and A. Richter. Angle–based indoor positioning system for open indoor environments. *Proceeding of Workshop on Positioning, Navigation and Communication (WPNC), Hannover, Germany*, 2009.

[3] M. Ciglar. A full-body gesture recognition system and its integration in the composition "3rd. pole". *ICMC*, 2008.

[4] M. Feldmeier and J. A. Paradiso. An interactive music environment for large groups with giveaway wireless motion sensors. *Computer Music Journal*, 31(1):50–67, Spring 2007.

[5] V. Pulkki and M. Karjalainen. Multichannel audio rendering using amplitude panning. *Signal Processing Magazine*, 25(3):118–122, 2008.

[6] F. Vogt, G. McCaig, M. A. Ali, and S. Fels. Tongue 'n' groove: An ultrasound based controller. *Proceedings of New Instruments for Musical Expression (NIME), Dublin, Ireland*, 2002.

Eobody3: a ready-to-use pre-mapped & multi-protocol sensor interface

Emmanuelle Gallin
Creation Department
Eowave
La Cure - 58110 Tintury - France
+33/(0)661 702 050
info@eowave.com

Marc Sirguy
Scientific Director
Eowave
La Cure - 58110 Tintury - France
+33/(0)661 702 050
info@eowave.com

ABSTRACT
Away from the DIY world of Arduino programmers, Eowave has been developing Eobody interfaces, a range of ready-to-use sensor interfaces designed for meta-instruments, music control, and interactive installations… With Eobody3, we wanted to create this missing link between the analogue and digital worlds, make it possible to control analogue devices with a digital device and vice versa: for example, to control a modular synthesizer with an IPad with no computer and vice versa. With its compatibility with USB, MIDI, OSC, CV and DMX protocols, Eobody3 is a two-way bridge between the analogue and digital worlds… This paper describes the challenge of designing a ready-to-use, pre-mapped, multi-protocol interface for all types of applications.

Keywords
Controller, Sensor, MIDI, USB, Computer Music, USB, OSC, CV, MIDI, DMX, A/D Converter, Interface.

1. INTRODUCTION
Developing a ready-to-use pre-mapped multi-protocol sensor interface is quite challenging. With Eobody2, we developed a range of sensor to USB interfaces commonly used for interactive installations, museography, live music and video performances, dance, but also for industrial and medical applications. With Eobody3, we wanted to go further. The popularized use of sensors in the communication and game industries has deeply influenced our control gestures, creating new reflex gestures and new control needs. Many would like to transpose these new control gestures to control music, synths or softwares, but there is no existing ready-to-use and multi-protocol bridge to create such interactions. Creating an interface that is dedicated to one particular application would be easy, but making the interface compatible with different protocols and existing products (different OS, softwares like Ableton Live, iPads or analogue synthesizers) and adaptable to many applications involves many other parameters.

2. DESIGNING A MULTI-PURPOSE PRE-MAPPED INTERFACE

2.1 Eobody philosophy
Since the creation of the first Eobody interface in 2002[1], Eowave has followed the same idea of making sensor control accessible for all. This means that anyone, with or without technical skills, would be able to use Eobody sensor systems. With the growing number of sensor controlled installations, performances, we've seen a lot of interest for these ready-to-use systems that enable artists to realize interactive creations without the assistance (and costs) of an engineer, or learning to program a microcontroller themselves.
On the other hand, engineers found that using these systems saved a significant amount of time, as they just had to set their own parameters to fit their needs. This specific 'plug & play for all" approach has a particular impact on the design process of our interface: 1) the technical side must be transparent to the user; 2) the design is focused on the way the interface will be used; 3) the accessible parameters are only "visible" setting parameters; 4) it imposes a wide compatibility with existing OS, softwares, MIDI devices and other hardware interfaces; 5) it requires different level of use: ready-to-use; internal parameter access via the editor; and Max programming; 6) it requires compatibility with other communication protocols.

2.2 From Eobody2 to Eobody3
Eobody2 was a USB MIDI sensor interface with 8 inputs, internal memory, and internal signal process[2]. After three years of existence, the customer response is still quite good, but we thought it was time to move towards Eobody3. We noted a recurring need among users to add different kinds of inputs and for a system that would be more open to protocols like OSC.
We used the Microchip PIC-32 MX microcontroller with a frequency of 80 MHz for 1,56 DMIPS/MHz. We also wanted the Eobody3 to be "evolutive", totally adaptable to future protocols and formats, with extensive number of inputs.

2.3 Question of times
Technologies evolve with time. iPhones and iPads with all sorts of musical Apps are now commonly used as control surfaces and create new needs. The idea of using an iPhone's surface like a "sensor" that would offer the potential to control the signal process of a CV or MIDI instrument needed to be explored.

3. EOBODY3 INTERNAL MODULAR ARCHITECTURE

The new Eobody3 project specifications involved the design of an internal modular architecture with a separated master motherboard, interchangeable daughter boards to host protocol format DSP, and a third board with inputs types (0-5V sensors, triggers, pedals and logic I/O) and outputs types (CV, PWM and digital I/O).

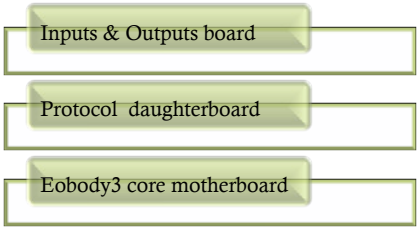


Figure 1 : Eobody3 modular architecture

3.1 Eobody3 core motherboard

Signal processes and mappings are made inside the Eobody3 core, so it requires no cpu from the computer. Different mappings are stored in the Eobody3 internal non-volatile memory, so it can be used without any computer when needed. The motherboard of Eobody3 is the brain of the system with a Microchip PIC32 programmed with pre-mappings for the different daughter boards. It hosts an updated version of the processing library Eobody2 Sensor Systems (ESS DSP) used in Eowave interactive devices. The role of the Eobody3 core motherboard is to process the raw data coming from daughter boards and to send them to the computer via USB or another chosen protocol. This modular architecture offers different advantages such as the ability to re-program the core for specific applications or to update the core with new mappings.

3.2. A core compatible with other communication protocols and I/O formats

The core is capable of interfacing through other communication protocols embedded on daughter boards, including OSC, MIDI, and DMX. On each daughter board, a Microchip 16 bit dsPIC receives data from the sensors, samples them at 30 kHz and transmits them to the core in high speed SPI. The choice of the compatible protocols was made considering the most commonly used protocols: MIDI is still widely used by musicians; OSC enables to use LAN cables on longer distance for live applications. Concerning DMX, there are many affordable DMX to USB converters on the market. Eobody3 is not just a DMX to USB converter but enables to connect a sensor directly to a DMX enabled device without a computer. The modular architecture also enables compatibility with new future protocols like Copperlan for example.

3.3 Input types

Inputs types can be chosen between 0-5V sensors (can be switched to 3,3V), triggers, pedals and logic I/O. Eobody3 can host from 8 to 32 inputs (in blocks of 8 inputs each).

The 0-5V input boards have a 1 pole low-pass filter and a buffer to accept different signals and filter data before the A/D conversion. Sensor input board dsPIC performs an average measure of incoming signal. Unlike the sensor input board, which measures continuous variations of the signal, trigger inputs enable to track impulses and attacks of the signal. A decoupling capacitor and rectifier diode at the input of the signal enable filtering of continuous components of the signal. The dsPIC tracks and keeps the highest values of the signal at each communication between the core and the input board. The connector does not have a 5V power on the tip of the TRS jack connector.

Pedals inputs have a similar architecture to the sensor input board, and allow switching automatically between Roland or Yamaha pedal format or footswitches.

Digital inputs/outputs provide logic 0/1 levels (0-5V).

3.4 Output types

Outputs types include CV, PWM and digital I/O.

Control voltage Output (CV) commonly used with analogue synthesizers, provides up to 8 x 0-10V outputs and 8 x gate (ON/OFF). These allow direct control over a synthesizer parameter with a sensor or by sending information from a computer. They also enable control of dimmers, light and any other voltage-controlled device. For modular synthesizers, Eobody3 core and Eobody3 compatible modules offer the possibility to patch directly outputs from a modular system to a computer or other digital devices. PWM enables to control motors and LED controllers. Sensors control the pulse width.

Digital inputs/outputs provide logic 0/1 levels (0-5V) that can be used to control relays.

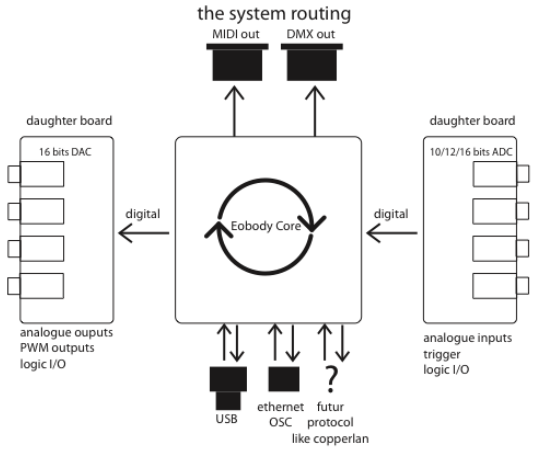


Figure 2: Eobody3 routing

4. PRE-ROUTING: FORMATTING MESSAGES TO HOST

This is one of the most important configuration parameters, since it determines which type of MIDI message the device is going to send in response to variations in a particular analogue input.

Eobody3 can generate 5 different message types:

CC: Control number change (control change) 7 bits or 12 bits (the 5 LSB bits are mapped on CC number + 64), PB: Pitch bend (variation in pitch) real 12 bits or mapped 14 bits, Monophonic aftertouch, Note on Trigger and Program change.

The analogue signal must correspond to an envelope changing with time and which has a maximum value. Users specify three parameters: program change sent, higher threshold, and lower threshold. Eobody3 analyses this envelope: once the envelope has reached the higher threshold, a MIDI program change or note message is generated. As long as the envelope remains above a

threshold, named lower threshold, the program change is maintained (no new MIDI message is sent). When the level falls beneath the lower threshold, Eobody3 is ready to receive a new message.

5. PRE-PROCESSING TOOLS

Data from sensors can be processed using the on-board pre-processing tools to get the best exploitable signal for a specific use. The Microchip PIC32 allows to have different processes according to the type of data, standard sensors (continuous controller), triggers (peak detection), simple logic or more complex algorithms for certain sensors like gyroscopes for example.

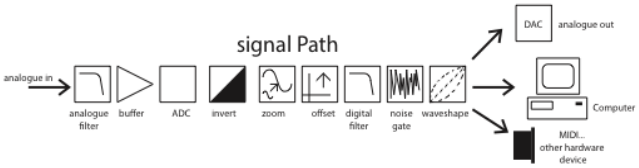


Figure 3: Signal path

Signal from analogue inputs goes thru analogue filters and buffers before A/D conversion. Then, pre-processing tools enable to reshape the signal with features like invert, zoom, offset, digital filter, noise gate, wave shaper. Pre-mapping settings per sensors are included.

5.1 Buffering the data flow

Common music softwares like Ableton Live or NI Reaktor, are only compatible with MIDI or OSC on Mac/PC OS. To have Eobody3 compatible with these, we were restrained to these protocols and their limitations in terms of update rate. With the Microchip PIC32, the update rate for all sensors has been reduced to less than 1ms, only delayed by the power of the computer to process the data flow. Real-time sensor interactions cannot accept the buffering techniques used with audio signal. Buffers need to be as low as possible to create an immediate perceptual sensation when a sensor is touched. To make this immediate perceptual interaction happen, the whole process needs to last less than 10ms (preferable 5). These 10ms include the audio process in the computer and the delay generated by the outgoing process of the soundcard. Tests made with USB show certain limitations with Windows7 as well as with OSX[3]: below 2ms, the midiin object in Max/MSP cannot process all incoming data and will "freeze" after a couple of seconds. Ethernet allows a larger data flow at once but the update rate is not faster. To limit the flow of incoming data, data from sensors are packed in Eobody3 and updated after a complete scanning cycle has been realized. This can be done with Ethernet as well as with USB.

5.2 Denoising data with gate and filters

Active sensors often generate noise. Eobody3 offers different types of processes to reduce noise. With Eobody2, an "analogue zoom" controlled the internal PGA (Programmable Gain Amplifier). With Eobody3, a pre-amp with analogue filters has replaced the PGA. Before the A/D conversion is done, the signal goes thru a low-pass filter and a unity gain buffer. This eliminates high frequency noises and allows sensors with high impedance outputs. After the conversion, sensors are processed as a 16-bit value, with the possibility of inversion, zoom and offset settings. Digital noise filtering is done by a 32-bit low-pass filter and a noise gate, which smoothes the signal. The noise gate threshold does not reduce the bit depth of the signal. If the signal moves in the range of the noise gate, no message is sent. This field enables to set the width of the range. A large range will be very effective against strong noise but will make the values less sensitive to a relevant change of the analogue signal. A threshold of 5 corresponds to 127 different possible values (i.e., the analogue has to change at least of 32 (from 4095 above or below its current position to be detected as a variation). A threshold of 11 corresponds to 2 bits, useful for switches or all on/off sensors.

5.3 Zoom: focusing on a part of the data

The digital zoom & offset parameters specify how the real range of an analogue input can be mapped on a 7-bit MIDI value. As a matter of fact, a sensor does not necessarily have a range equal to the reference voltage of the Analogue to Digital Converter (ADC). A custom-scaled zoom has been implemented on the digital value to take advantage of the 12-bit resolution of the A/D converter. First, the voltage reference has to be set to the largest range among the sensors connected to the unit. Then, specifying a window size and an offset can set a sensor's range within the 12-bit dynamic. The selected range can then be converted into 7-bit MIDI data without greatly increasing the quantification step. Another application of the digital zoom & offset parameters might be to reduce or adapt the range of the sensor to control a specific range of a filter parameter in a plug-in, for example.

5.4 Transfer curves new tool

Eobody3 has a new pre-processing tool called curve (for transfer curve). This new wave shaping feature allows for changing the curve of a signal. For example, the linear response of a volume pedal can be changed to a logarithmic or exponential response.

5.5 Pre-mapping per sensor type

Pre-mapping is the art to translate the raw data coming from the stimulated sensor into a signal that will be immediately perceived as a cause to effect. « Making these mapping choices, it turns out, is anything but trivial. Indeed, designing an interactive system is somewhat of a paradox. The system should have components (dance input, musical output) that are obviously autonomous, but which, at the same time, must show a degree of cause and effect that creates a *perceptual interaction*[4]. » To create this perceptual interaction, each sensor needs a dedicated mapping depending on its affectation. With the design of a multi-purpose interface such as Eobody3, the challenge consisted in the pre-mapping of all sensor types for all applications. For this, all types of sensors have been studied to get the most versatile pre-mapping, which usually differs from (but includes) the most frequent pre-mapping. We integrated on-board calculation of complex pre-mapping with algorithms for accelerometers and gyroscopes as well as a Kalman filter. While such mappings were only possible with a Max patch or through coding an Arduino, Eobody3 gyroscopes or accelerometers are now ready-to-use and are fully compatible with MIDI applications like Ableton Live. Depending on the treatment they receive, accelerometers can provide two kinds of information: force and gravity. To measure acceleration, the DC component of the signal is filtered whilst the AC component is filtered to get gravity information. The gyroscope can track angular movement calculated from the rotation information. This calculation

requires an accurate timing, though outsourced treatments generate timing errors due to USB and computer variable delays. The generated drift needs a constant initialisation of the sensors when it returns at its zero position. Information from accelerometer and gyroscope together will provide more precision.
In the editor, selecting the sensor type will automatically call the pre-mapping for this sensor-type. Pre-maped values can be re-shaped using the pre-possessing tools to get even more customized data.

5.6 Pre-mapping for triggers

With Eobody2, we had a lot of demands for using triggers as well as sensors. This was not possible, as velocity processing would have required a more powerful microcontroller. With the Microchip PIC32, we have integrated this velocity process. Trigger input level can be adjusted with a sensibility parameter. Threshold sets the beginning of the trigger analysis and peak detection. Release time sets the time before a new detection is processed. Cross talk enables Eobody3 to cancel unwanted messages when two trigger pads are very closed one another for example. In this case, input 1 can cancel a message coming from input 2 if those are simultaneously triggered. The velocity response curve can be modified with a wave shaper. With triggers, USB connectivity allows response with less than 2ms delay.

6. EOBODY3 EDITOR

Like Eobody2, Eobody3 offers three levels of use, from ready-to-use to the opened max file[5]. In most cases, users only need the ready-to-use level. With applications like Ableton Live or other MIDI softwares, Eobody3 will be automatically recognized as an audio device and can control any available MIDI parameters of the software.
At a second level, when more parameters edition is needed, Eobody3 editor gives access to editable pre-mapping configuration and other pre-processing tools. All settings can be stored in Eobody3 non-volatile memory. The third level offers the Max5 editable file of the editor.

Figure 4: Eobody3 core editor

7. A New bridge

7.1 Control your modular system and other CV synths with your iPhone or iPad

While communication and game industries are releasing new powerful control tools, the world of musical interfaces seems far beyond. While many musical Apps appear on the market[6], many would like to use iPad or iPhone control technology as new controllers. But Apps are specific to their own environment and cannot be used as generic controllers. With Eobody3, we propose to make a real bridge between these worlds and offer musicians the possibility to use their iPads or iPhones to control any CV or MIDI gears without a computer thru the CV ouputs connected to a USB cable or via a LAN network with a wireless router.

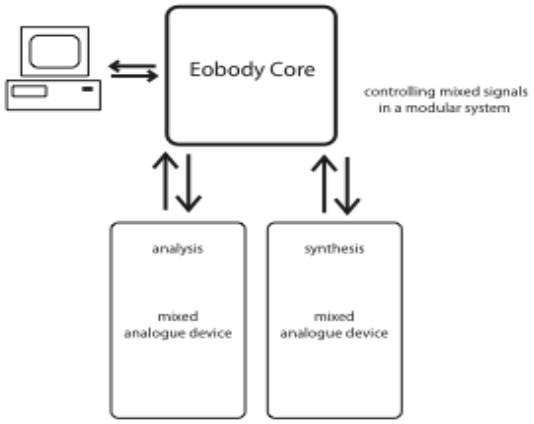


Figure 5: Controlling mixed signals in a modular system

7.2 Eobody3 compatible modules for modular systems

We've extended this idea with the development of dedicated analogue modules that comply with Eobody3 standard. With these, it will be possible to use multiple analogue filters controlled by a Max application. This opens the door to complex computer analysis and analogue re-synthesis, which give the users a unique possibility to get the best result from both analogue and digital worlds.

8. CONCLUSION

The idea of Eobody3 is to offer a new ready-to-use sensor interface that can be used for controlling music, video, but also lights, analogue synthesizers, relays, motors… It offers anyone the possibility to design his own interactive installations, meta-instruments or sensor-based control surface without any background in programming and electronics.

[1] E. Fléty, M. Sirguy : Eobody : a Follow-up to AtoMIC Pro's Technology, Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montreal, Canada. NIME03-225.
[2]Eobody2 specifications http://www.eowave.com/downloads/pdf/eobody2_usb8sensorbox_manual.pdf
[3] Tests made with intel core duo.
[4] Joseph Butch Rovan, Robert Wechsler and Frieder Weiß : *Seine hohle Form: Artistic Collaboration in an Interactive Dance and Music Performance Environment*, Crossings.
[5] For other Eowave interfaces, EoMessage V 1.0, is a dedicated Max object developed by Eowave to control any ESS DSP based interfaces, EoMessage can be used with Eobody2 OEM board, Eobody2 HF, Eomono. It will easily enable to use ESS DSP based systems within a Max/MSP environment.
[6] NAMM 2011, January 13-16, Anaheim CA, USA.

Eye Tapping: How to Beat Out an Accurate Rhythm using Eye Movements

Rasmus Bååth*
Lund University
Cognitive Studies
Kungshuset, Lundagård
222 22 Lund, Sweden
rasmus.baath@lucs.lu.se

Thomas Strandberg
Lund University
Cognitive Studies
Kungshuset, Lundagård
222 22 Lund, Sweden
thomas.strandberg@lucs.lu.se

Christian Balkenius
Lund University
Cognitive Studies
Kungshuset, Lundagård
222 22 Lund, Sweden
christian.balkenius@lucs.lu.se

ABSTRACT

The aim of this study was to investigate how well subjects beat out a rhythm using eye movements and to establish the most accurate method of doing this. Eighteen subjects participated in an experiment were five different methods were evaluated. A fixation based method was found to be the most accurate. All subjects were able to synchronize their eye movements with a given beat but the accuracy was much lower than usually found in finger tapping studies. Many parts of the body are used to make music but so far, with a few exceptions, the eyes have been silent. The research presented here provides guidelines for implementing eye controlled musical interfaces. Such interfaces would enable performers and artists to use eye movement for musical expression and would open up new, exiting possibilities.

Keywords

Rhythm, Eye tracking, Sensorimotor synchronization, Eye tapping

1. INTRODUCTION

Humans have the ability to synchronize movements to an external rhythm. This is an unique ability not found among any other mammal, not even among the greater apes. Recently it has been shown that parrots and cockatoos have a limited ability to entrain to music [2]. Still it can not compare with the richness of human rhythmical expression whether it is dance or music making. It is not strange then that *sensorimotor synchronization*, the rhythmic coordination of perception and action, is an active field of research [11].

While most research is done on tasks such as finger tapping, eye movements and rhythm is an unexplored area. This lack of research might be because of the technical difficulties in measuring eye movements. Unobtrusive devices for measuring eye movements, so called *eye trackers,* have been available for many years. To accurately measure the timing of eye movements the temporal resolution of an eye tracker should be high however, and this is a problem as many eye trackers have had and still has relatively low temporal resolution (< 250 Hz). Nowadays there are commercial high-speed eye trackers available, some with a temporal resolution of 500 Hz or more, and resolution should not be a limitation anymore. There exists a few examples of using eye movements to control music [6, 4, 8] but none of them describe the use of eye movements to trigger sounds in a rhythmical fashion. A recent study [5] used eye movements to generate hand clap sounds in time with a metronome.

When finger tapping a rhythm, or when drumming, it is obvious when the actual strike to be synchronized with the beat is made. It occurs when the finger or drum stick hits a surface. When beating out a rhythm using the gaze and an eye tracker there is no surface to strike and it is not obvious when the actual strike is made.

When beating out a rhythm using gaze, henceforth called *eye tapping*, where is the "strike" felt? One obvious alternative is to use eye blinks as the triggering method but even if only the actual eye movements are used there are still many alternatives. Two types of eye movements that could be used for eye tapping are *fixations* and *saccades*. A fixation occurs when the gaze maintains the focus on a single location; saccades are the fast eye movements made between fixations. An example of how to trigger a sound by an eye movement would be to use the fixation onset, another example would be to trigger a sound in the middle of a saccade.

The aim of this study was twofold. The first aim was to investigate if, and how well, it would be possible to beat out a rhythm using eye movements. The second aim was to establish the most accurate method of eye tapping. There are four main reasons why it is interesting to investigate how to best control rhythm with the eyes and how well it can be done:

1. **To make it possible to use eye moments for musical composition.** Many parts of the body are used to make music; the mouth and lungs control wind instruments, fingers and arms control string instruments, and legs and feet are used when drumming but so far, with a few exceptions [6, 4], the eyes have been silent. To enable performers and artists to use eye movement for musical expression would open up new, exiting possibilities.

2. **To enable people with physical disabilities to make music.** Eye tracking is already used by people with physical disabilities to interact with computers. If more was known about how to control rhythms with eye movements it would make it possible for them to enjoy new opportunities for musical expression.

3. **To learn more about sensorimotor synchronization.** The literature on sensorimotor synchronization

*Corresponding author.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

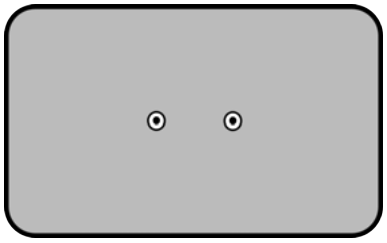


Figure 1: The display shown to the subjects during the experiment. The distance between the two fixation points was either 10 or 20 visual degrees.

has been preoccupied with studying finger tapping and few other means of striking a beat has been tried. Eye movements and finger movements might have different properties and our knowledge about sensorimotor synchronization in humans would not be complete without accounting for eye movements.

4. **To enable the use of rhythmic control in gaze controlled interfaces.** Knowledge about how to best control rhythm with eye movements could not only be used in interfaces for music composition but rhythm could more generally be used as a input variable in gaze controlled interfaces. The most obvious application being computer games as the game mechanics in many games are rhythm based.

A goal with the study was also to generate useful guidelines for what properties of rhythmic eye movements that needs to be considered when implementing an eye movement based instrument.

2. METHOD

18 subjects (11 male) were recruited from the student population of Lund, Sweden. All were volunteers and no payment was given. Their age ranged from 19 to 50 with a mean age of 26. Twelve reported to have had musical training and on average a subject had 7 years of musical training.

Subjects were seated in front of a tower-mounted, SMI Hi-Speed eye tracker with a temporal resolution of 500 Hz. A chin rest was used to constrain subjects' head movements and maintain an eye-to-screen distance of 67 cm. The screen used as stimuli display was a Samsung SyncMaster 245T, a 52×33 cm large LCD monitor with a resolution of 1920×1200 px.

The task given to the subjects was to let their gaze alternate between two fixation points in time with a beat (see fig. 1). The beat was given by a sequence of 50 msec square wave beeps of 440 Hz with a fixed inter-onset interval (IOI). The beeps were played to the subjects through a pair of full sized head phones (Philips SHP2500). A subject was given 16 session, where each session consisted of 20 sec. of beat following and 10 sec. of rest. The sessions were kept short in order to avoid fatigue of the subjects. Two factors were varied in the sessions, tempo and the span of the fixation points, and each session included one combination of factors. Tempo was either 60 or 120 bpm, corresponding to an IOI of 1.0 and 0.5 sec. The span of the fixation points was either 10 or 20 visual degrees. Each factor combination was used in four sessions but the order of the combinations was randomized for each subject. Both visual and auditory stimulus was presented using Matlab[1] with the Psychophysics Toolbox extensions [7] and eye tracking data was recorded using iView X 2.5[2].

After collecting the eye tracking data five different methods of eye tapping were implemented and applied on the gaze position data. This yielded the tap onsets that would have been present if the eye tapping methods had been used during the sessions. As the fixation points were placed on a horizontal line only the x-coordinates of the gaze position were used. The five methods to generate taps were:

1. When crossing the midline. This generates a tap every time the midline between the fixation points is crossed, but only after 100 msec have been spent on one side of the midline.

2. At the beginning of a saccade. This generates a tap when leaving a fixation point after having looked at it for more than 100 msec.

3. At the end of a saccade. This generates a tap when arriving at a fixation point after having crossed the midline.

4. At maximum saccade velocity. This generates a tap at the velocity peak when saccading between the two fixation points.

5. When fixating. This generates a tap every time a fixation is made after having crossed the midline. A gaze point is defined as a fixation if the point-to-point velocity is below $20°/s$ [12]. Note that this eye tapping method does not use the actual position of the fixation points.

Each beep onset was then paired with the closest tap onset generated by each of the five methods (see fig. 2). Beep onsets from the three first sec of each session were disregarded as no cue was given subjects when a session was about to start.

One variable of interest is beep-to-tap asynchronies, that is, the difference between the beep onsets and the closest taps of the methods. Here a negative value would indicate that the tap onset occurred before the beep onset and a positive value would indicate that the tap onset occured after the beep onset. This will show if subjects tend to eye tap before or after the beat. This is not a good measure of performance however. If the beep onsets are [1.0, 2.0, 3.0, 4.0] and the corresponding taps generated by one of the methods above are [0.8, 1.9, 3.0, 4.3] this gives the asynchronies [-0.2, -0.1, 0.0, 0.3] but a mean asynchrony of 0.0. A better performance measure is to take the absolute value of the asynchronies, [0.2, 0.1, 0.0, 0.3], which then gives a mean *absolute* asynchrony of 0.15. This is the main variable of interest and it shows how well the different methods, and the subjects, performed.

Processing of the eye tracking data was done using the Ruby programming language[3] and statistical analyses were conducted using the R environment [9]. The raw eye tracking data and the script used to generate eye taps are available for download at `http://www.sumsar.net/files/eye_tapping_experiment1.7z` .

3. RESULT

3.1 Performance of the Tapping Methods

Table 1 summarizes the result of the experiment. All tapping methods had a mean absolute asynchrony in the range of 130-190 msec. The mean direction of the asynchronies was negative for all methods, that is, all methods tended to

[1] `http://www.mathworks.com/`

[2] `http://www.smivision.com/`

[3] `http://www.ruby-lang.org/`

beeps	midline	beginning	end	velocity	fixation
3	2.74	2.72	2.77	2.72	2.96
4	3.93	3.91	3.97	3.93	3.99
5	4.77	4.75	4.80	4.77	4.83
6	5.93	5.92	5.96	5.92	6.00
7	6.83	6.81	6.86	6.83	6.89

Figure 2: An excerpt from session one of subject one with beep onsets (IOI = 1.0 sec) and the corresponding tap onsets generated by the five eye tapping methods.

	Mean	SD	Abs mean	Abs SD
(1) Midline	-71	185	163	112
(2) Saccade beg.	-58	203	173	121
(3) Saccade end	-22	261	185	185
(4) Max. velocity	-61	202	169	125
(5) Fixation	-49	175	137	119

Table 1: Mean and SD in msec. of the asynchrony and absolute asynchrony and of the five tapping methods for the 18 subjects.

yield taps that occurred before the beep onset. A repeated measures ANOVA was used to test for absolute asynchrony differences between the five tapping methods. Absolute asynchrony differed significantly across the five methods, $F(4, 17) = 5.76, p < 0.001$. Fisher's LSD test showed that method (5), to generate a tap when fixating, had a significantly lower mean absolute asynchrony than the other methods ($p < 0.05$). This is also visible in figure 3 .

3.2 Performance of the Subjects

As the fixation based tapping method was significantly better that the other four methods it was used in the subsequent analyses of the performance of the subjects. The mean asynchrony and the mean absolute asynchrony of the subjects are shown in figure 4. The mean absolute asynchrony over all subjects was 137 msec ($SD = 55$) and a one sample t-test showed that it differed significantly from zero ($p < 0.001$, 95% CI[110, 165]). All subjects, except one, had a negative mean asynchrony. A one sample t-test showed that mean asynchrony over all subjects differed significantly from zero ($M = -49.6$ msec, $p < 0.01$, 95% CI[-77, -22]). This is comparable to the negative mean

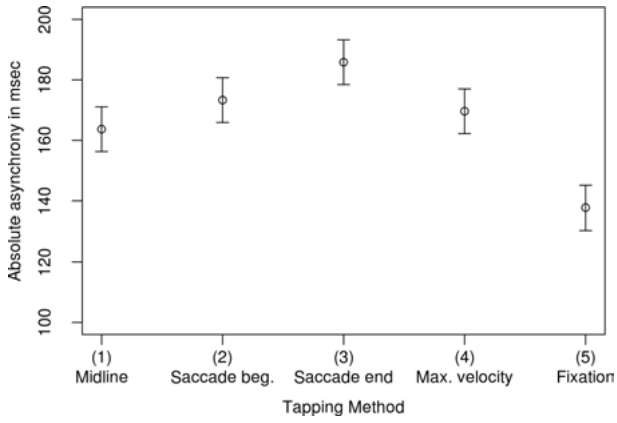


Figure 3: Mean absolute asynchrony, in msec., of the 18 subject as a function of tapping method. The error bars shows the standard error given by the ANOVA.

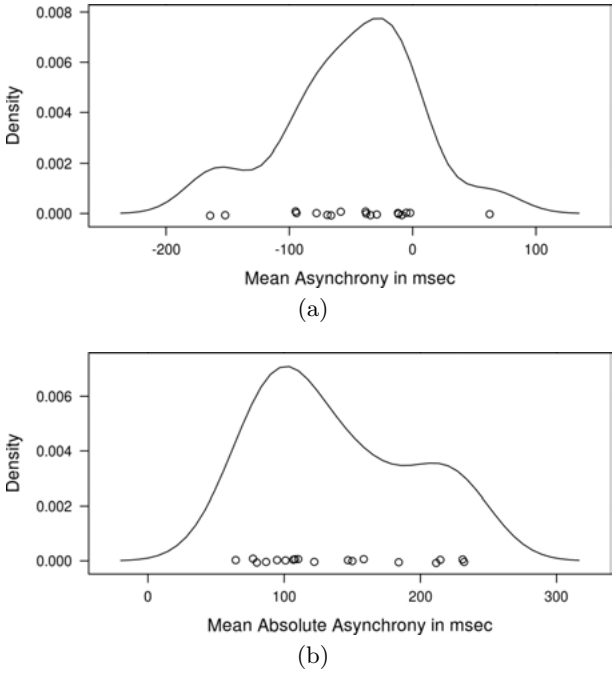


Figure 4: Mean asynchrony (a) and mean absolute asynchrony (b) in msec of the fixation tapping method for each subject. The points show the means of the individual subjects and the line shows the density created using a gaussian kernel.

asynchrony of 20–80 msec frequently found in finger tapping tasks when no auditory feedback is given [1]. Another finding in the finger tapping literature is that musicians perform better in finger tapping tasks than non-musicians [3, 11]. No significant correlation between reported number of years of musical training and mean absolute asynchrony was however found (Pearson's product-moment correlation, $r = -0.035$, $p = 0.89$). A negative correlation was found between mean asynchrony and mean absolute asynchrony ($r = -0.67$, $p < 0.01$), that is, bad performance is related to a tendency to eye tap to early. A positive correlation between the standard deviation of the mean asynchrony and the mean absolute asynchrony ($r = 0.89$, $p < 0.001$) indicate that low performing subjects are also less consistent in their timing.

4. DISCUSSION

The most accurate method of triggering rhythmical sounds from eye movements is by using fixation onsets (5) as the fixation based method had a significantly lower mean absolute asynchrony compared to the other methods. When building an eye controlled instrument accuracy might not be the only criteria when choosing an eye tapping method. Another criteria might be ease of implementation and if this is important the second best method, the midline based (1), might be a better choice.

All subjects managed the task of synchronizing their eye movements to a given beat and the conclusion is that it is possible to beat out a rhythm using eye movements. What is evident is that eye tapping differs from finger tapping and drumming in that subjects are more inaccurate [10]. The mean error in this study was 137 msec which is a quite noticeable error when beating out a rhythm. If implementing an eye controlled instrument one should be prepared for that the rhythmic accuracy of the users of your instrument might be quite low. Another difference from finger tapping

studies is that no effect of musical training was found. It would be interesting to see if, and how much, training of eye tapping over an extended period can improve accuracy.

In this study there was a strong tendency of negative mean asynchrony. When finger tapping the negative mean asynchrony is known to decrease or even disappear when auditive feedback is given [11]. An interesting continuation of this study would then be to conduct an experiment where fixation based eye tapping is used to trigger taps and to compare the results with the finger tapping litterature. What also should be done is to use eye tapping to create new musical interfaces to allow performers and artists to use eye movements to create and perform music.

5. ACKNOWLEDGEMENT

This study was conducted at the Humanities Laboratory at Lund University[4]. The authors gratefully acknowledges support from the Linnaeus environment Thinking in Time: Cognition, Communication and Learning, financed by the Swedish Research Council, grant no. 349-2007-8695.

6. REFERENCES

[1] G. Aschersleben. Temporal Control of Movements in Sensorimotor Synchronization. *Brain and Cognition*, 48(1):66–79, 2002.

[2] W. T. Fitch. Biology of music: another one bites the dust. *Current biology : CB*, 19(10):R403–4, May 2009.

[3] M. Frank, J. Mates, T. Radil, K. Beck, and E. Pöppel. Finger tapping in musicians and nonmusicians. *International Journal of psychophysiology*, 11(3):277–279, 1991.

[4] A. Hornof and L. Sato. EyeMusic: making music with the eyes. *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04)*, pages 3–5, 2004.

[5] A. Hornof and K. Vessey. The sound of one eye clapping: Tapping an accurate rhythm with eye movements. In *Proceedings of the 55nd Annual Meeting of the Human Factors and Ergonomics Society, to appear.*, 2011.

[6] J. Kim, G. Schiemer, and T. Narushima. Oculog: playing with eye movements. In *Proceedings of the 7th international conference on New interfaces for musical expression*, pages 50–55. ACM, 2007.

[7] M. Kleiner, D. Brainard, and D. Pelli. What's new in Psychtoolbox-3. In *Perception 36 ECVP Abstract Supplement.*, 2007.

[8] A. Polli. Active vision: Controlling sound with eye movements. *Leonardo*, 32(5):405–411, 1999.

[9] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010.

[10] B. Repp. Rate limits in sensorimotor synchronization with auditory and visual sequences: The synchronization threshold and the benefits and costs of interval subdivision. *Journal of Motor Behavior*, 35(4):355–370, 2003.

[11] B. H. Repp. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12(6):969, 2005.

[12] D. Salvucci and J. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 71–78. ACM, 2000.

[4] `http://www.humlab.lu.se/`

MelodyMorph: A Reconfigurable Musical Instrument

Eric Rosenbaum
Lifelong Kindergarten Group
MIT Media Lab
Cambridge, MA
ericr@media.mit.edu

ABSTRACT

I present MelodyMorph, a reconfigurable musical instrument designed with a focus on melodic improvisation. It is designed for a touch-screen interface, and allows the user to create "bells" which can be tapped to play a note, and dragged around on a pannable and zoomable canvas. Colors, textures and shapes of the bells represent pitch and timbre properties. "Recorder bells" can store and play back performances. Users can construct instruments that are modifiable as they play, and build up complex melodies hierarchically from simple parts.

Keywords

Melody, improvisation, representation, multi-touch, iPad

1. INTRODUCTION

For the improviser, an instrument's interface creates a landscape of possibilities: from a particular gesture, some gestures are "nearby" and others are far; there is a set of familiar pathways that are easy to traverse; and there is a tradition of idioms to fall back on. This landscape can change over time, but slowly, because the physical configuration of instruments is typically fixed: the order of the piano keys or the tuning and fretting of a guitar do not change as you play.

What would it be like to radically reconfigure the interface to your instrument as you play it? A reconfigurable instrument could have the potential to create new possibilities for improvisers, and open up a new creative space. By "reconfigurable," I am referring to the ability to change the spatial mapping between gestures and sounds in real-time as you play the instrument. Imagine an exploded piano, with individual keys that you can position anywhere you want them.

Some affordances of traditional instruments will be lost with such an instrument, of course. The familiar pathways, if any, will be only temporary ones. And there is no tradition of idioms for a reconfigurable instrument (at least, not yet).

But other affordances unique to the new genre of reconfigurable instruments could be gained. For example, because the notes can be positioned anywhere, the proximity of gestures to each other is completely redefinable. For example, the leaps in an angular melody might make it difficult to play on a piano, but could be made easy on a reconfigurable instrument. Another new affordance is customizability. A reconfigurable instrument could be rearranged to suit the needs of a particular player, a particular composition, or even a particular moment in time.

I present MelodyMorph, a reconfigurable instrument designed with a focus on melodic improvisation. It is designed for a multi-touch screen interface. A palette allows the user to create "bells," graphical representations of individual notes that can be played by tapping and moved by dragging. Special "recorder bells" can store and play back sequences of bells. A pannable and zoomable canvas enables the creation of large melodic landscapes. Canvases can be saved and reloaded for later use. My initial investigations suggest that this interface is usable, fun, and ripe with new possibilities.

2. RELATED WORK

The inspiration for MelodyMorph comes in part from a set of educational manipulatives called Montessori Bells. They are small metal bells on wooden stands, identical in appearance but varying in pitch. The bells are used with young children to pose puzzles about pitch (which two are the same?) or melody (can you construct this tune?). Jeanne Bamberger has investigated children's thought processes as they worked with the bells and invented their own notations that act as instructions for other children to play them [1]. Drawing on this work, one of the initial concepts motivating the MelodyMorph project is the idea that people could use it to create something that acts as both a "notation," visually representing the structure of a melody, and simultaneously as an instrument that lets you expressively play that melody.

There is relatively little existing work in the area of reconfigurable instruments. Tangible controllers such as Audiopad [2] and reacTable [3] allow the user to control sound synthesis and sequencing by positioning and manipulating pucks on a table. Unlike MelodyMorph, these controllers focus on controlling parameters, rather than constructing melodies. The reacTable-based scoreTable [4] focuses on melody construction, but it uses a sequencer metaphor. Sequencers in general have fixed mappings, so they are not reconfigurable by my definition. Pin & Play & Perform [5] enables the ad-hoc construction of instruments, by pinning dials, sliders and buttons onto a conductive substrate. The focus of that project is more on controlling musical parameters than playing melodies. Similarly, the Spinner project [6] enables users to freely map physical dials onto GUI controllers for musical parameters.

3. MELODYMORPH DESIGN

3.1 Palette

The palette is shown at the top of the screen (see figure 1), and enables the user to quickly create any number of bells. It shows one octave at a time of a chromatic scale. Buttons at the sides slide the palette up or down an octave, giving a total of three octaves. An instrument switcher at the top of the palette makes three different instrument timbres available.

3.2 Bells

Each bell can be tapped to play its corresponding note. An animation showing it briefly increasing in size and then shrinking provides feedback that it is playing. A bell can be dragged by pressing it near the center (causing it to play), and then dragging beyond its edge. A bell can be deleted by dragging it back to the palette.

The bells have different shapes to represent the different instrument timbres: circles for bass, squares for piano, and triangles for vibraphone. Their colors indicate different pitches. The range of twelve pitch classes is mapped onto the cycle of hues, resulting in a rainbow of notes. The lower octave is

shown with darker colors, and the upper octave is shown with lower saturation, for paler colors. Additionally, visual textures are used to differentiate the "function" of each note within the key of the root note shown in the palette. Chord tones (1, 5, and 8, in the chromatic scale) are shown with a solid color. Other tones in the major scale are shown with a horizontal band (3, 6, 10 and 12). Tones outside the major scale have vertical stripes (2, 4, 7, 9, and 11).

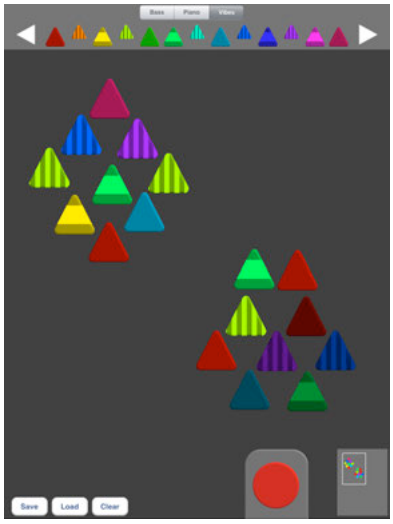


Figure 1: The MelodyMorph Interface

3.3 Expression

When each bell is tapped, the force of the tap is estimated using data from an accelerometer in the axis perpendicular to the screen. This value determines the loudness of the resulting note. Additionally, if a bell is held down while it is playing, an accelerometer in the plane of the screen is used to determine pitch bend. This enables control of pitch, by wiggling or tilting the whole device, over a range from subtle pitch vibrato to whammy effects.

3.4 Canvas

The bells inhabit a canvas much larger than the screen. The canvas can by panned by dragging anywhere there is not a bell. It can be zoomed in and out with a pinch gesture. Zooming way out gives a view of the entire composition. Zooming way in makes the bells larger and easier to tap. A small "mini-map" in the corner of the screen shows the entire canvas, with colored dots representing the bells and a box showing the current screen view.

3.5 Recorder bells

A recorder widget at the bottom of the screen can be tapped to toggle on recording. Any bells that are played are recorded into a special "recorder bell," which appears as soon as recording is toggled off again. This recorder bell behaves in many ways like a regular bell. Tapping it triggers the playback of its recorded sequence. A tap during playback stops the sequence. The recorded sequence is notated on the surface of the bell as a sequence of colored dots, with pitch on the vertical axis, and normalized time on the horizontal axis (see the white and gray objects in figure 2). During playback, the notation is animated, with each note appearing as it plays. The recording process includes recording the playback of other recorder bells, enabling complex melodies to be built up in layers.

3.6 Saving

A canvas with its configuration of bells and recorder bells can be saved as a file, along with a thumbnail image of the screen, for later reuse.

3.7 Implementation

MelodyMorph has been implemented on an iPad, under iOS 3.2. The functionality is built primarily using OpenFrameworks, with some Objective-C for user interface elements.

4. SCENARIOS

Here I will provide two examples of ways people might use the MelodyMorph system.

4.1 Kalimba Making

A simple use case (and one the author enjoyed early in the development process) involves constructing a small spatial arrangement of notes that are consonant with each other, and playing patterns on them with the fingers or thumbs (see e.g. figure 1). The result is a bit like a Kalimba, or thumb piano, except that it is completely customizable, and can even be modified during a performance.

A refinement of this technique involves constructing two or more kalimba patterns based on related chords, and switching between them to create a more complex improvised structure.

4.2 Hierarchical Melody Construction

A more elaborate use case involves using the recorder bells to build up a complex structure out of simpler elements.

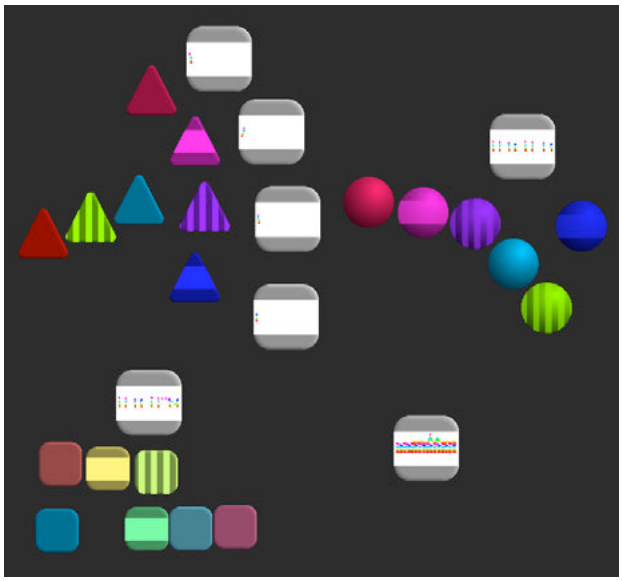


Figure 2: A more complex MelodyMorph construction

Figure 2 shows a melodic structure built up in layers. At the top left, a minor triad is shown along with four different upper structure notes (representing a standard descending "line cliché"). The four recorder bells next to these notes contain four note minor chords constructed with each of them. These four chords were then recorded in a sequence, twice through, resulting in the recorder bell at the top right. Below it, bass notes are shown in a pattern that matches the descending line cliché, along with two additional chord tones. This was used to play a bass line along with the chord sequences, resulting in the recorder bell at bottom left. Below that, a cluster of piano notes was constructed for the purpose of improvising a melody along with the accompaniment created so far. An improvisation over

four repetitions of the accompaniment was recorded into the recorder bell shown at the bottom right.

5. EVALUATION AND FUTURE WORK

A priority for continued work on MelodyMorph is to carry out some initial user studies, consisting of careful observations of users with various degrees of musical training as they play and improvise with the interface. These studies will likely reveal bugs and usability issues that can then be resolved.

One problem with MelodyMorph is that the touch screen provides no tactile feedback, making it difficult to tap the bells accurately, especially when zoomed out. It's not clear how best to provide this feedback. A tangible version of MelodyMorph could provide such feedback, but would only be feasible if the individual bells (each with sensing and communication on board) could be made cheaply enough that a large number could be fabricated.

Another problem with MelodyMorph is in synchronization. Because it does not use a sequencer metaphor, there is no fixed time base. It can be difficult to accurately time a melody when "overdubbing" on to a recorder bell. It may become desirable to add a toggle-able metronome to provide a tempo reference.

I am considering several features to add to the system. An annotation system would allow users to draw in freehand on the canvas, so they could do things like label melodic sections, decorate their instruments, and create flowcharts showing how to play larger melodic structures.

An additional palette section for "transformation" elements would contain objects that can be applied to bells and recorder bells, effecting musical transformations such as transposition, harmonization, timbre changes, etc.

The system of colors, textures, and shapes to represent pitch, function and timbre will be evaluated for its intuitiveness and possibly redesigned. Similarly the notation system for recorder bells may need to be redesigned.

MIDI or OSC output would enable MelodyMorph to send control data to other synthesizers, creating much more flexibility in possible timbres.

A networking feature would enable multiple devices to share data, such as the ability to pass groups of bells and recorder bells between devices.

6. REFERENCES

[1] Jeanne Bamberger. 1995. The Mind behind the Musical Ear: How Children Develop Musical Intelligence. Harvard University Press, Cambridge.

[2] James Patten, Ben Recht, and Hiroshi Ishii. 2002. Audiopad: a tag-based interface for musical performance. In Proceedings of the 2002 conference on New interfaces for musical expression (NIME '02), Eoin Brazil (Ed.). National University of Singapore, Singapore, Singapore, 1-6.

[3] Sergi Jorda. 2003. Sonigraphical instruments: from FMOL to the reacTable. In Proceedings of the 2003 conference on New interfaces for musical expression (NIME '03). National University of Singapore, Singapore, Singapore, 70-76.

[4] Sergi Jorda and Marcos Alonso. 2006. Mary had a little scoreTable* or the reacTable* goes melodic. In Proceedings of the 2006 conference on New interfaces for musical expression (NIME '06). IRCAM, Centre Pompidou, Paris, France, France, 208-211.

[5] John Bowers and Nicolas Villar. 2006. Creating ad hoc instruments with Pin&Play&Perform. In Proceedings of the 2006 conference on New interfaces for musical expression (NIME '06). IRCAM, Centre Pompidou, Paris, France, France, 234-239.

[6] Shigeru Kobayashi and Masayuki Akamatsu. 2005. Spinner: a simple approach to reconfigurable user interfaces. In Proceedings of the 2005 conference on New interfaces for musical expression (NIME '05). National University of Singapore, Singapore, Singapore, 208-211.

The Flo)(ps: Negotiating Between Habitual and Explorative Gestures

Karmen Franinovic
Interaction Design
Zurich University of the Arts
Ausstellungsstrasse 60
Zurich, Switzerland
karmen.franinovic@zhdk.ch

ABSTRACT

The perceived affordances of an everyday object guide its user toward habitual movements and experiences. Physical actions that are not immediately associated with established body techniques often remain neglected. Can sound activate those potentials for action that remain latent in the physicality of an object? How can the exploration of underused and unusual bodily movements be fostered? This paper presents *the Flo)(ps* project, a series of interactive sounding glasses, which aim to foster social interaction by means of habitual and explorative sonic gestures within everyday contexts. We discuss the design process and the qualitative evaluation of collaborative and individual user experience. The results show that social interaction and personal use require different ways of transitioning from habitual to explorative gestures, and point toward possible solutions to be further explored.

Keywords

sonic interaction design, gesture, habit, exploration

1. INTRODUCTION

What we know how to do strongly affects what we do, what we perceive and what we are willing to do [**?**]. Given a glass and a pitcher filled with water, we will most likely pour the water into the glass, although its shape suggests many other movements, such as rolling and throwing the object. The latter actions, however, are neglected because we do not associate them with a range of past experiences of using a glass. Abandoning such functionality of an everyday object in the name of exploration and play may be suitable within contexts in which social interaction is at the focus, such as in bars or clubs. How can these existing social experiences be extended by through explorative sonic actions?

As witnessed by our past research [**?**, **?**], the unusual objects within public setting, such as new musical instruments, attract and engage the user in an explorative discovery of its potential for action. In contrast, the use of an everyday object results in the most obvious and expected gestures which are often exploited for engaging interaction. For example, in the *musicBottles* project, the user can play a song by removing the cork from the bottle, providing her with the sensation of freeing music from the object [**?**]. Using the bottle as a sound container, the user expression is limited to acting with its cork as an on-off switch. A more explorative sonic interaction with an everyday object can be found in the *Audio Shaker* project where sounds can be mixed by interacting with an ordinary looking cocktail shaker [**?**]. Users can speak into the object to record sounds, close and shake it to re-mix them and then literally, pour out the sound mix. The potential of the object and its affordances challenge the user's preconceptions about the purpose of the cocktail shaker through an unusual sonic feedback. While the continuity of *Audio Shaker's* feedback allows for a more explorative interaction then the discrete responses of the *musicBottles*, both interfaces engage habitual actions such as opening the bottle or pouring the sound. But how can an everyday sounding object guide the user toward the space of unusual and explorative gestures?

Figure 1: Luminous Flo)(ps glasses with Martini cocktail drinks.

The exploration with novel musical interfaces can be engaged by shaping the coupling between action, sound and object. The material aspects of an interface, such as its shape, weight or texture, afford an energy transfer between the body and the instrument [**?**]. When interaction with a novel instruments is designed around such physical qualities and without reference to another known object, the user has to explore its potential in order to learn how to generate sounds (for example, see project such as [**?**]). Such process of learning and discovery is enabled by coupling action to sonic feedback in expected and natural ways as well as in unusual and novel relationships [**?**]. The individual repertoires of expected couplings are defined through shared, culturally encoded movements and shaped by specific personal skills, such as knowing how to skate or to play an instrument [**?**]. This existing bodily knowledge may serve as a starting point

NIME'11, 30 May–1 June 2011, Oslo, Norway.

for unusual and novel embodied experience.

2. FLO)(PS CASE STUDY

The Flo)(ps are a set of interactive glasses designed to explore habitual and explorative sound gestures with everyday objects, and their impact on social and personal interaction within an existing situation (See Figure **??**). These glasses sonically respond to habitual actions such as cheering and drinking, but are also activated when certain unusual gestures are performed. Different glasses can establish connection among each other if synchronously moved in a similar manner. Their connectedness is manifested through sonic and light responses which signal to the users that they are affecting the behavior of others glasses. The goal of such performative connectedness is to make strangers play with each other through an everyday object in an embodied, dance-like way.

The concept for connected glasses was one of the results of basic design research in which we explored the relationship between action and sound in the use of mechanical kitchen tools [**?**]. Our exploration was directed toward revealing the existing action-sound relationships and informing the design of new computational artifacts that produce sound. The results of this research were then applied within the context a project on intimacy in public space, resulting in an ecology of interactive cups that can engage strangers in non-verbal communication [**?**].

A simplified version of *The Flo)(ps* interface was used in psychological experiments on the emotional impact of sound during the performance pouring task [**?**]. While within this controlled setting only one action-sound relationship was studied, that of dropping the invisible ice cubes out of the glass, the full version of the interface presented in this paper consists of eight additional action-sound couplings, and added sensing and actuating elements. Also, the focus of this research is the design of the habitual and unusual sonic gestures and the evaluation of their impact on individual and social interaction.

2.1 Design

There is a number of multitouch products that exemplify the potential of interactive technology within bar setting [**?**, **?**]. In addition, the robotic glasses presented in [**?**] showed how the autonomous behavior of an everyday object can engage social interaction. Similarity with the technical and design solutions of *The Flo)(ps* can be found in [**?**] where the glasses are used within the telepresence application. Other colleagues explored the sonification of kitchen actions including pouring and stirring and argued for the continuity of sound feedback as a key element for engaging embodied interaction [**?**].

2.1.1 Technical Aspects

The shape of *the Flo)(ps* glass was designed comfortably fit in the hand and intuitively allow for a range of movements such as twirling. The form was modeled in 3D software and extruded with 3D printer (Dimension BST 768) in non-toxic ABS plastic. The lower part of the glass contains the electronics: the Arduino BT board with its shield hosting an RGB LED and three sensors, and the lithium batteries. An analog devices ADXL 320 3-axis MEMS accelerometer captures movements performed with the glass. A piezo-microphone is glued to the shell of the glass in order to capture surface interactions such as the impacts and the scratching. Finally, the Capacitive Sensor Board - AD7746 Breakout measures the level of the water and communicates when the glass is filled with liquid. The sensor data is sent through Bluetooth connection to a remote computer where it is processed and the real-time sound is synthesized in Cycling'74 Max/MSP. The sound is played back through speakers positioned in the proximity of the glass and embedded in the bar where the drinks are served.

2.1.2 Action-Sound Couplings

The design of action-sound couplings took place through sonic bodystorming where we explored sonic gestures using different objects and materials (See Figure 2). We continued testing our decisions tacitly throughout the design process. In addition to individual use of the object, we also explored the interaction between two people such as throwing the sound toward someone. This helped us decide which habitual and non-habitual gestures should be identified from sensor data and how these should be mapped to different sounds.

Figure 2: Sonic Bodystorming: Probing sound concepts by the use of digital and analog means.

In total eight different gestures were extracted from sensor data. The habitual gestures included filling the glass with liquid, raising the glass, stirring the liquid, drinking and toasting, and the unusual gestures comprised twirling, moving the glass very slowly and shaking the glass. Habitual gestures generated sound of liquids such as pouring or splashing while strange movements opened up unexpected sonic spaces such as the sound of the wind or the rain. The movement of the glass continuously changed the qualities of the sound in order to give the user the feeling of an "ecologic experience", in the sense of cause and effect behavior found in physical phenomena. For example, tilting the glass would make some virtual water come out and than stop until the user inclined the glass more in order to pour out the remaining water.

2.1.3 Experience Design

The glasses respond to the user only when they are full, and otherwise sit quietly waiting to be filled with the liquid. Once filled, the glasses start to pulsate luminously and emanate the sound of water drops, each in its own rhythm: faster and irregular, slow and in patterns or slow and regular. Different responses aim to communicate specific identity of each glass - one is energetic and nervous, one is slow and relax, one is determined and clear. Their behavior is intended to attract the visitors: as they approach the glass the volume of the sound increases, and once the glass is grasped it fades out. The habitual actions, which are expected to be firstly performed, activate cartoonified liquid sounds or "sounds that caricature some aspects of the events while omitting others." ([**?**] pg 14). Starting from such existing action-sound repertoires, the user is guided into new movement spaces. For example, twirling the glass activates the sound of the wind. The wind sound grows louder and

more complex if the user continues to twirl the glass.

Figure 3: The Flo)(ps setup at Oboro gallery with loudspeakers embedded inside the bar.

The *the Flo)(ps* glasses can affect each others sonic properties when gestures are performed simultaneously. When the movement is synchronous, same light and sound patterns are displayed. The aim of using the light feedback is to establish visual link when users are too far apart from each other. The connective sound responses would become stronger as the users move in the domain of non-habitual movement with the glass. In this way, the users may influence each other's movements through the sonic and light response of the glass. The goal of connected behavior of *The Flo)(ps* is to allow the users to collectively perform and "to dance" with each other encouraged by the response of interactive glasses. For the video presentation of the project please visit the website [**?**].

2.2 Evaluation

Considering that the main goal of the project is to connect strangers through performative acts with everyday objects, the evaluation aimed to reveal the social potential of the system and to gain understanding into individual experience of using *the Flo)(ps* glasses within public setting.

2.2.1 Context

The artifacts were exhibited at the Oboro center in Montreal, Canada over three-week period at the International Design Biennale, St. Etienne, France over four-week period. The Oboro exhibition allowed the users to drink beverages from the glasses whereas this was not possible at St. Etienne Biennale due to the large number of approximately 85.000 visitors. Thus, the main evaluation was undertaken by analyzing data collected during the Oboro exhibition.

In this installation, each of the three glasses was associated with an area of the bar below which the speaker was located and chairs were used to keep the visitor's interaction bounded to that designated bar area (Figure 3). Although the exhibition was opened every day, the drinks were served in the late afternoon and evening of each weekend, at the exhibition opening, special events such as Journees de la Culture and special organized visits (e.g. a group of students). These events lasted from two to five hours.

2.2.2 Methods

As we have shown in the past, the range of social experiences that emerge within public installations in large part cannot be predicted [**?**]. Thus, we preferred to qualitatively evaluate user's natural interaction with the system without any previous instructions, rather than basing evaluation on a specific task which could be quantitatively measured. In order to collect data about the user experience, we deployed questionnaires and direct observation including participant observation, design-adopted video ethnography and the informal interviews [**?**]. These methods were applied sequentially in order to avoid guiding user experience through questions. Firstly, the visitors interactions were video recorded; then participatory observation combined with informal interviews took place; and finally, the questionnaires were provided after the groups of visitors finished interacting. Thus, the data collected included more than six hours of video recordings, seventeen filled questionnaires and notes from the participatory observation and interviews with participants during the installation.

2.3 Analysis and Findings

The average interaction with the glasses took fifteen minutes, although many visitors spend more time within the installation while chatting with friends and drinking from *the Flo)(ps*. The participants statements quoted in the text below are accessible here [**?**].

2.3.1 Social Interaction

Overall, the findings about the social dynamics emerging around the objects proved to be best defined from the analysis of the video material and the insights gained through informal interviews. A number of patterns were seen to emerge and some of the social phenomena that were noted include:

- *Mirroring and Synchronizing:* Participants were observed to mirror each other's movements, especially when someone discovered a new sonic behavior, as if learning from each other.

- *Non-verbal Communication:* Overall, the glasses succeeded in enhancing non-verbal communication. One visitor wrote that: "It is socially engaging because you don't have to talk to connect with strangers since you are already linked by the sound you are making and also the gestures". Another visitor described sound as "an extension of body language";

- *Collaborative Music Performance* was observed, as groups of three participants aimed to collaboratively compose sounds. This often led them to ignore the programmed sound and light connections as they focused on musical improvisation;

- *Simple Sonic Play* such as creating sound of clinging glasses by toasting was repeatedly performed. Participants appeared to enjoy the simplicity and predictability of the direct feedback. However, this sometimes appeared to limit further exploration of interactivity;

- *Curiosity and Discussion:* Participants proposed different interpretations of the objects and explanations for their use. Discussions about using the object to simplify the work of the waiters or as a seduction tool arose;

- *Ambient Display:* During the play and performance with the objects, participants would stop to talk to someone while enjoying their drinks. The glasses would fade into the background until the user's attention was drawn back to their responses.

These observations show that the installation forged interaction between strangers, by engaging them with sonic and light gestures. The light feedback appeared to have a stronger connection effect than the sound response which often appeared to be too complex to interpret. The use of light feedback rhythmically varying in color and luminosity had an important role in establishing contact and was necessary when the surrounding soundscape grew louder.

Figure 4: The empty Flo)(ps used as a musical instrument, without reference to habitual gestures.

2.3.2 Individual Experience

The subjective aspects of the experience were best described within the questionnaires and in participants reflections collected during the informal interviews. The following findings emerged:

- *Expressive Solo Performance:* Most visitors experienced the object as an expressive instrument that engaged playful interaction. However, when they interacted alone, the rhythm of the performance slowed down and they were able to more carefully explore the behavior of the object;
- *Exploring the Unusual:* Participants found it difficult to link unusual sounds such as those of the wind to the glass. However, they were satisfied that the sound continuously responded to their gestures and created new unusual experience. One visitor wrote: "swirling it in a slightly less habitual and functional manner, it opens up an unusual sonic space. The splashing sound seems to gain in resonance. Soon after a deep howling, evocative of a storm, becomes amplified." [**?**];
- *Limited by Habits:* Few participants stated that certain assumptions about what should be done with the glass affected their experience. One participant wrote: "I was more focused on solitary interaction. I guess I assumed that all that could be done with the glasses could be done alone.". Small deviations from habitual events, such as toasting with glasses of different materials (i.e. plastic glass with the sound of crystal one), were well accepted by the participants, but may have limited their explorations;
- *More Dynamics:* Visitors who played for a short time period said that the sounds should evolve or change more often. The sounds did not evolve sufficiently if only habitual gestures were performed;
- *Introspection and Intimacy:* Many visitors who were alone in the installation used the glass as a kind of relaxation tool. They were observed to stare at the drink being illuminated by the light or to slowly twirl the glass while listening to the sound of the rain or the wind. One visitor wrote: "They remind me of candles. It would be cool if they reacted to the stress in your palms.";
- *Strangeness:* Several comments suggested that the sounds confer the sense of strangeness. Participants associated sounds to "an imaginary chemistry lab", "stalagmite space", "a damp basement" and "outer space". Others however linked them to personal memories such as "Playing in bath as a kid" or sensations like "the sounds make me feel like I am underwater".

When performing individually, the participants were more attentive to sensuous responses of the system. They engaged more deeply with exploring the transitions between habitual and unusual gestures while providing different interpretations for the sonic and luminous behavior of the glass. The results show that the distribution of the attention of the user is the key to moving between different types of interaction offered by the system. Although the user interaction cannot be predicted, the spaces in-between the solitary and collective performance and between habitual and unusual gestures should be better choreographed.

2.3.3 Discussion

The presence of liquid in the glasses showed to be highly significant in affecting and constraining the way in which users interact with the glasses, and the ways in which they perceive them. In the setting where the glasses were activated even when no liquid was in them, the glass was immediately interpreted as a musical instrument, a toy or a magical device (See Figure 4). This raised a question of how to balance the expressivity and the existing functionality of an everyday object.

Figure 5: A girl listening to the sounds.

The usability issues emerged when too many people were present in the location as the participants could not hear the sounds well (See Figure 5). Also those who left the bar area that was linked to their glass had difficulty hearing and associating the sound source to the actions they performed. However, they continued to play by relying on the light feedback. The integration of the speaker in the body of the glass is necessary in order to improve usability and to conduct exploration within real-world context such as a dance club or a caffe.

Some visitors expressed desire for simpler and clearer sound responses. The clarity of interaction may be improved by reducing the number of gestures and by using simple gestures such as rhythmical patterns or large movements such as raising the glass, both of which showed to be preferred by the participants due to their clarity. However, prudence is required as reducing the temporal evolution of sonic feedback to direct responses only may lead to on-off behavior which could quickly bore the user. In fact, those participants who had interacted alone desired to hear more complex sonic behaviors. Thus, one solution to be tested is

to apply simple behaviors when more people are using the glasses and more complex ones when a solo interaction take place.

3. CONCLUSION

We have described the design and the qualitative evaluation of the use of *The Flo)(ps* interactive glasses that aim to stimulate connectedness among strangers through sonic movement. The goal was to explore the space in-between the habitual and unusual action-sound interactions with an everyday object. We observed that using a familiar object such as a cocktail glass may facilitate the first exposure to its interactivity, but it may also limit the exploration of its behavior due to the assumptions about their use. Our findings show that such objects can engage users in non-verbal communication, especially if the action-sound relationships are simple. Strategies for the transition between habitual and experimental actions still remain to be explored. In this direction, the next steps for *The Flo)(ps* project will focus on the more abstract sonic feedback for habitual actions in order to break the sonic expectations of the user and facilitate unusual gestural interaction.

Strangeness may be the key to exploring the boundary between the familiar and the unknown gestures, as witnessed by this reflection of one of the *The Flo)(ps* users: "As I become immersed in my experimentation with the drinking glasses, their familiarity gradually becomes odd to me, in the way a word can gradually acquire a strangeness if we repeat it over and over again. This turn from familiarity to estrangement allows for a rediscovery" [**?**]. It is our hope this project raised questions and awareness that such playful and embodied reflection can be stimulated and sustained through novel sonic experiences within our everyday contexts.

4. ACKNOWLEDGMENTS

The author wishes to thank Yon Visell for software development, Martin Peach for electronics advice, Fabienne Meyer and Thomas Tobler for fabrication support. This research was supported by the European Commission project CLOSED FP6- NEST-PATH no. 29085. and the Hexagram research Interstices Lab, Montreal.

5. REFERENCES

[1] H. R. Bernard. *Research Methods in Anthropology: Qualitative and Quantitative Approaches, Fourth Edition*. Altamira Press, 2005.

[2] C. Cadoz. Instrumental gesture and musical composition. In *Proceedings of the International Computer Music Conference*, pages 1–12. International Computer Music Association, 1988.

[3] H. Chung, C.-H. J. Lee, and T. Selker. Lover's cups: drinking interfaces as new communication channels. In *CHI '06 extended abstracts on Human factors in computing systems*, CHI '06, pages 375–380, New York, NY, USA, 2006. ACM.

[4] K. Franinović. Flo)(ps website, http://zero-th.org/flops.html.

[5] K. Franinović. Enactive encounters in the city. In P. Beesley, S. Hirosue, J. Ruxton, M. Traenkle, and C. Turner, editors, *Responsive Architectures: Subtle Technologies*. Riverside Architectural Press, 2006.

[6] K. Franinović. Basic interaction design for sonic artefacts in everyday contexts. In *Focused - projects and methods of current design research*. Swiss Design Network Symposium, 2008.

[7] K. Franinović and Y. Visell. New musical interfaces in context: Sonic interaction design in the urban setting. In *NIME '07: Proceedings of the 2007 conference on New interfaces for musical expression*, 2007.

[8] W. W. Gaver. How do we hear in the world? explorations in ecological acoustics. *Ecological Psychology*, 5(4):285–313, 1993.

[9] M. Hauenstein and T. Jenkin. Audio shaker - http://www.nurons.net/audioshaker/about.htm.

[10] intactive. Intactive multitouch bar, http://intactive.de/.

[11] H. Ishii, A. Mazalek, and J. Lee. Bottles as a minimal interface to access digital information. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2001.

[12] A. R. Jensenius. *Action-Sound: Developing Methods and Tools to Study Music-Related Body Movement*. PhD thesis, University of Oslo, Department of Musicology, 2007.

[13] M.-H. Lemaire. Protected: Blurred and playful intersections - karmen franinovic's flo)(ps. *Wi: Journal for Mobile Media*, March 2009.

[14] G. Lemaitre, O. Houix, K. Franinović, Y. Visell, and P. Susini. The flops glass: A device to study emotional reactions arising from sonic interactions. In *Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference*, 2009.

[15] M. Mauss. *Sociology and Psychology: Essays*, chapter Body Techniques. Trans. by Ben Brewster. Routledge and Kegan Paul, 1979.

[16] Mindstorm. Mindstorm ibar, http://mindstorm.com/products/ibar.

[17] A. Noe. *Action in Perception*. MIT Press, 2004.

[18] F. Rey, M. Leidi, and F. Mondada. Interactive Mobile Robotic Drinking Glasses. In H. Asama, H. Kurokawa, J. Ota, and K. Sekiyama, editors, *Distributed Autonomous Robotic Systems 8*, pages 543–551, Berlin Heidelberg, 2009. Springer.

[19] D. Rocchesso and P. Polotti. Designing continuous multisensory interaction. In *Proc. of Sonic Interaction Design workshop at Computer-human interaction conference*, Firenze, 2008.

[20] E. Singer. Sonic banana: A novel bend-sensor-based midi controller. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression, Montreal*, 2003.

Wekinating 000000Swan: Using Machine Learning to Create and Control Complex Artistic Systems

Margaret Schedel
Stony Brook University
Stony Brook, NY
margaret.schedel@stonybrook.edu

Phoenix Perry
NYU Poly
New York, NY
phoenix@areyoudevoted.com

Rebecca Fiebrink
Princeton University
Princeton, NJ
fiebrink@princeton.edu

ABSTRACT
In this paper we discuss how the band 000000Swan uses machine learning to parse complex sensor data and create intricate artistic systems for live performance. Using the Wekinator software for interactive machine learning, we have created discrete and continuous models for controlling audio and visual environments using human gestures sensed by a commercially-available sensor bow and the Microsoft Kinect. In particular, we have employed machine learning to quickly and easily prototype complex relationships between performer gesture and performative outcome.

Keywords
Wekinator, K-Bow, Machine Learning, Interactive, Multimedia, Kinect, Motion-Tracking, Bow Articulation, Animation

1. INTRODUCTION
Obsessed with electronics, rare birds, myth, Native American art, pagan ritual, fetish, punk, and tribal percussion, 000000Swan is an experiment in performing process and interaction. We create high-impact, hard-to-predict events beyond the realm of normal expectations, performing on a variety of electronic instruments including keyboards, a JazzMutant Lemur, a Zeta cello with a sensor bow, and a Kinect. We are able to quickly create interactive audio and visuals by harnessing the power of machine learning with Wekinator. In this paper, we discuss how we created the interactive audio and visual elements for the song *Monster*.

2. HARDWARE AND SOFTWARE

2.1 Wekinator
The Wekinator [2][3] is a freely available software environment designed to facilitate the interactive application of supervised learning to real-time problem domains, including music.[1] Supervised learning algorithms are a family of machine learning algorithms capable of using a training dataset to produce a model (see, e.g., [1]). This model can be understood as a function capable of producing some output value (e.g., a gesture label, such as "staccato") from some input value (e.g., a feature vector computed from sensor bow outputs). The training set consists of a set of example input-output pairs (e.g., each pair might consist of a single feature vector and the true gesture label that should be applied to that feature vector). Supervised learning has been an effective tool for building models in many problem domains in which labeled training data is available, but where the relationship between features and labels is too complex to specify explicitly in code. Musical gesture identification and mapping creation are two such domains in which prior work has found supervised learning to be useful (e.g., [6][9][12]).

The Wekinator provides a graphical user interface for collecting and editing training data, training learning algorithms, and running trained models to produce outputs from inputs in real-time. Users create training examples by specifying the target output (e.g., gesture class) in the GUI and demonstrating the corresponding gesture or other input signal; features are extracted from the user's input and saved with the target value. Wekinator includes implementations of standard discrete classification algorithms (k-nearest neighbor, decision trees, support vector machines, and AdaBoost.M1), as well as multilayer perceptron neural networks for regression. Users are able to interactively change algorithms, algorithm parameters, and selected features. Significantly, users are also able to influence model behaviors by adding, deleting, and editing the training examples. Compared to other machine learning tools, the Wekinator was designed to more explicitly support rapid, iterative model design through interactive changes to the training dataset [3].

Once a user has created a model by training a chosen algorithm, (s)he can run the model to produce predicted outputs for incoming feature vectors that are extracted in real-time. In our bow gesture classification system, for example, the user can execute different types of bow gestures using the K-Bow and observe the model's predicted output over time.

2.2 Kinect
The Kinect is a hands-free accessory for Microsoft's Xbox 360. It uses an RGB camera in combination with a depth sensor and multi-array microphone to enable users to interact with video games without a physical controller.[2] It was released in the USA in November of 2010 and was quickly hacked to enable units to send data directly to computers via the USB port. In our performance, we use the depth data as input to supervised learning models created by the Wekinator, allowing us to use body movement to control and trigger both audio and video.

2.3 K-Bow
The K-Bow is the first commercially-developed, mass-produced sensor bow for string players [7]. It contains 1) a three-axis accelerometer located inside the frog, which senses tilt and acceleration of the bow in space; 2) a grip sensor that perceives changes in the grip pressure and surface area of the

[1] http://wekinator.cs.princeton.edu/

[2] http://www.xbox.com/en-US/kinect

cellist's bow hand; 3) an angle-sensitive pressure sensor located at the junction between the bow hair and the frog, which measures changes in the tension of the bow hair; and 4) an infrared detector inside the frog, which measures the bow position and angle relative to a circuit board and IR emitter mounted under the fingerboard.

The K-Bow ships with a software suite, K-Apps, which receives sensor values from the bow. This software provides a GUI interface for sensor calibration and sends sensor values to other software programs via OSC or MIDI. We use data from the K-Apps as input to several Wekinator models to control and trigger audio and visuals in our performance.

2.4 Audio and Visual Software

Ableton Live is a Digital Audio Workstation optimized for live performance.[3] Using data from the keyboards, Lemur, Zeta Cello, and Wekinator we are able to control audio processing, launch samples and loops, as well as play software synthesizers while simultaneously controlling synthesis parameters. For example, the lead singer might be playing keyboards while data from the K-Bow adjusts the distortion on the patch she is playing.

Unity is an integrated graphical environment for creating 3D games and animations.[4] Its game engine runs on multiple platforms including Windows and OS X, a web plug-in, iDevices, and most commercial game consoles. Using data from the Wekinator we are able to control an interactive game environment, launching visuals, changing colors and camera angles, and creating generative graphics such as particle systems to create visuals for *Monster*.

2.5 Data Flow

Our data flow is illustrated in Figure 1. K-Apps receives K-Bow sensor outputs and forwards them to a standalone feature extractor, which extracts features (e.g., minima and maxima, first- and second-order difference) and sends them to Wekinator via OSC [11]. Simultaneously, rudimentary features are extracted from the Kinect to roughly describe the 3D location of the human performers, and these are sent to the Wekinator as well. Certain Wekinator models are trained to create and control Ableton Live sounds in response to features extracted from the K-Bow and/or Kinect, and other Wekinator models are trained to drive aspects of the Unity game engine.

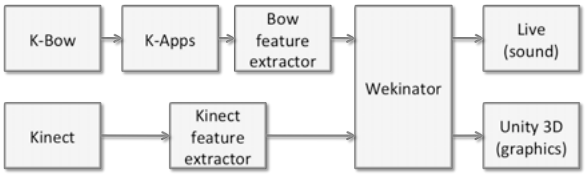


Figure 1. Data Flow for 000000Swan

3. DISCRETE CLASSIFICATION OF BOW ARTICULATIONS

3.1 Prior Work using K-Bow and Wekinator

Prior research has shown the discrimination of string bow strokes and articulations to be tractable using sensor bows and machine learning (e.g., [9][12]), though this work has not studied the production of classifiers that were later used in live performance. In our own prior work, we used the Wekinator to create eight bow stroke classifiers for the 000000Swan cellist using the K-Bow. For example, our articulation model classifies seven standard bow articulations (see [4]): legato (smooth and connected), marcato (onsets emphasized and slightly detached), spiccato (enunciated and percussive), riccocet (bouncing, rapid notes), battuto (struck with the wood of the bow), hooked (re-articulation of notes without a change in bow direction), and tremolo (rapid alternation of up-bows and down-bows). The classifiers were constructed to identify articulations played on any string of the cello and to be reasonably robust to changes in horizontal and vertical bow position (i.e., frog, middle, tip; sul tasto, sul ponticello), bow pressure, and bow speed. The articulation classifier was the most complex model that we built, and it achieved a 98.8% cross-validation accuracy and a subjective quality rating by the cellist of "9" out of "10."

3.2 Discrete Classification Experience with 000000Swan and Wekinator

In performance we found we needed a way for the cellist to trigger discrete events, much like a button on the Lemur. During difficult vocal passages for the lead singer, we decided it was much more important to focus on the vocal line, versus attempting to both sing and trigger, therefore the cellist needed to be able to trigger samples. We tried using particular notes on the cello, but the unique notes for triggering stood out from the rest of the cello line. Triggering from bow position did not give satisfactory results; the only way to make it consistent left only two possible triggers at the very tip and directly at the frog. By using the seven identifiable articulations from the Wekinator in combination with string information and extreme bow position (i.e. frog, tip, ponticello, and sul tasto), we were able to recognize 112 unique triggers which we use for multiple songs in a set.

3.3 Discrete Classification in *Monster*

One way we use the bow articulation triggers in *Monster* is to change the color of the visualization. The bass line is consistent throughout each verse, but the first verse uses a legato bowing to produce a purple visualization, and the second uses marcato to create white particles.

We also use bow articulations to trigger samples; almost inaudible riccocets, tremolos, and batuttos on the A and D string in the ponticello and sul tasto positions enable the cellist to trigger 12 discrete audio samples varying in length from 0.2 seconds to a minute-long ambient sweep towards the end of the piece. This ability to add elements during the performance is important to us; an integral element of the 000000Swan aesthetic is to make each live show unique.

4. CONTINUOUS CONTROL USING KINECT

4.1 Prior Work in Continuous Control with Wekinator

The Wekinator has previously been used by other composers to create interactive systems in which performers' gestures continuously control sound synthesis parameters [8][10]. In those compositions, as in components of our own work, composers used the Wekinator to prototype, refine, and perform with many-to-many mapping functions built from neural networks. Unlike prior compositions, we have combined continuous and discrete control mechanisms, and we engage gestures of multiple performers to control both audio and visuals.

[3] http://www.ableton.com/
[4] http://unity3d.com/

4.2 Continuous Control Experience with 000000Swan and Wekinator

Our lead singer has a dance background, and we often work with aerialists. We wanted a way to use body movement to control aspects of the performance. We programmed several tracking systems in Max/MSP/Jitter/SoftVNS, but we were unhappy with the results. Either the mapping from gesture was too direct and uninteresting, or else the tracking was not robust. In addition, the system ran very slowly. Using the Wekinator circumvented these problems. Since the Kinect sends formatted vision tracking data directly into Wekinator the environment is very responsive. Unity runs directly on the GPU so we are able to create complex visuals without taking up too much of the CPU, leaving more power for Ableton.

With Wekinator we are able to quickly train models to drive sound and visuals in response to gestures performed in front of the Kinect. We can train models for specific spaces by creating training examples in the venue before the performance. For example, we may use the downward motion of the aerial dancer to manipulate the EQ on a synthesizer patch in Live. The range of the dancer's height changes depending on the elevation of the rigging. We only need to give Wekinator two training examples—one at the top of the dancer's range, mapped to a narrow EQ of 2, and one at the bottom, mapped to a wide EQ of 18—to recalibrate the height-EQ model for a new venue. This is a simple mapping, but we also use the Wekinator for many-to-many mappings as discussed in the next section.

We also use the Kinect to track the musicians' movement to influence sound and visuals. Previous tracking systems were very dependent on costumes and lighting; using the depth sensor from the Kinect, we have eliminated lighting as a variable. Since the Wekinator is so easy to train we can create models in multiple costumes, making our performances more robust.

4.3 Continuous Control in *Monster*

In *Monster*, we use a particle generator to create interactive visuals. One layer on top of the particle generator is a spiral of triangle shapes. The position of the triangles is controlled by the position of the right arm of the lead singer. The Kinect is able to track this variable through the entire field of the camera. Using neural networks to create continuous mappings from arm position to triangle position allows the visuals to respond dynamically to gradual changes in the singer's movement.

We also use body position to control camera parameters in Unity. Using a set of five Wekinator models, we are able to create a many-to-many mapping between performer gesture and Unity's camera focus, angle, and 3D position. The same position features are used to drive seven of Live's processing parameters. Performers thus affect the visuals and audio in complex and dynamic ways that which would be difficult, if not impossible to code by hand.

We train these models in the venue using four types of training examples: 0) Standing close, cello playing arco, left hand high on the strings 1) standing close, cello playing pizzicato, left hand low on the strings 2) standing far apart, cello playing arco 3) lead singer crouching, cellist leaning backward 4) lead singer with arms in the air, cellist kneeling. We know basically what visuals and audio processing will result from these position states, but we do not know how the "in between" states will react. We know generally what will happen, but sometimes the results surprise us. For example, if the lead singer is crouching and the cellist is kneeling, the visual state may be somewhere between (3) and (4), but we don't know for sure until we experiment with the trained models. This poses a creative challenge; we want the system to be predictable and reproducible, while remaining engaging. This type of mapping is also rewarding in that, by creating "meta-sensors" driven by the actions of both performers, each member has her own role in shaping the collective experience.

5. DISCUSSION

5.1 Advantages of Machine Learning

000000Swan is extremely pleased with the Wekinator. Previous interactive systems we developed were not robust over multiple venues and costumes, we found it difficult to program complex results, and we felt we were spending more time coding than working on the music and visuals. With the Wekinator we are able to take complex streams of information from multiple controllers and quickly program audio and visual responses. We use both concrete classifiers as triggers and continuous classifiers to transform between states.

We see five real advantages to using interactive machine learning in our multimedia performance:

1) **Efficiency in design:** We no longer have to parse complex sensor information ourselves. Instead of trying to understand eight variables coming in from the K-BOW every 10ms, and thousands of IR points coming from the Kinect every 33ms, we can think about the bigger picture and let Wekinator handle the details.

2) **Customizability:** The Wekinator is fast to train; we are no longer anxious about how our system will respond in different venues. We simply train the program in each setting, in costume. We create models in our dress rehearsal but retain the output response.

3) **Supporting complex performer-performer interactions:** The Wekinator does not distinguish between the types of data coming in; therefore, we can track multiple sensors at the same time to create "meta-sensors." This augments the interaction between the performers.

4) **Supporting complex mapping strategies:** We can create both discrete triggers and continuous control in the same program, and Wekinator's neural networks create complex, interpolating systems with many-to-many mappings without a lot of programming.

5) **Rapid prototyping:** We can re-train and experiment quickly with different models for the same sensors to control the sound and visual environment.

To some extent, the practical advantages that machine learning offers in creating customizable, complex mappings without explicit programming are inherent to the use of a generative mapping strategy (e.g., see [5]). In our experiences, these benefits are also contingent on the ability to rapidly create, explore and change the machine learning models. A less interactive system that did not allow us to experiment with changing training examples, that took a long time to train, or that made it difficult to quickly test models by running them on real-time inputs would be significantly less useful to our work.

5.2 Disadvantages of the Wekinator

Although we are happy with the Wekinator, there are some disadvantages that we have had to work around.

1) There is no explicit support for triggering. In Max/MSP/Jitter it is trivial to create a trigger. With the Wekinator you must go through a secondary router in order to create a trigger.

2) The Wekinator's OSC output messages aren't customizable in format. We therefore rely on a third-party

routing software (OSCulator) to translate them into the correct format for Ableton.

3) There isn't an easy way to "turn off" the output of selected Wekinator models. The easiest way to program Ableton for control by an external OSC process is to click on the parameter to control (e.g., volume) and move the controller (and only that controller). However, because Wekinator's models all continuously output values simultaneously, OSCulator was also used for this function.

In order to streamline our workflow, we are working with the creator of Wekinator to improve the software by allowing triggering and greater control over its OSC output behavior.

5.3 Other control strategies in *Monster*

We don't use Wekinator for all of the controller data in *Monster*. We a keyboard to send traditional MIDI in order to play synth pads, and we use a LEMUR to send OSC directly to Ableton, using sliders to control the volume of the singers, electronic sound and cello and buttons to launch the song, and to trigger samples. For one-to-one mappings, such as the horizontal bow position mapping to distortion on the synth pad we bypass the Wekinator and simply use the OSC data from K-Apps.

6. CONCLUSION

We have summarized our use of machine learning techniques in driving sound and graphics in our live interactive performance, *Monster*. Our use of these techniques builds on a large foundation of prior work that has demonstrated the feasibility of applying machine learning to gesture analysis and mapping creation. Through the use of the Wekinator software, we have been able to put these techniques into practice in our own work.

Although the use of the Wekinator software has required us to create several extra software modules for feature extraction, the most significant impact machine learning has had on our work is the reduction in the need to write code and the expansion of control possibilities available to us. As a result, more of our development and composition time has been devoted to exploration of these possibilities, and our attention has been more focused on cultivating the creative and artistic qualities of our work.

7. REFERENCES

[1] Bishop, C. M. 2007. *Pattern Recognition and Machine Learning*, 2nd ed. Springer.

[2] Fiebrink, R. 2011. *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance.* PhD thesis, Princeton University.

[3] Fiebrink, R., Trueman, D., and Cook, P. R. 2009. "A meta-instrument for interactive, on-the-fly machine learning." In *Proc. Intl. Conf. on New Interfaces for Musical Expression (NIME).*

[4] Flesch, C. 2000. *The Art of Violin Playing: Book One*. Carl Fischer, New York, NY, USA.

[5] Hunt, A., and M. M. Wanderley. 2002. "Mapping performer parameters to synthesis engines." *Organised Sound*, 7(2): 97–108.

[6] Lee, M., A Freed, and D. Wessel. 1992. "Neural networks for simultaneous classification and parameter estimation in musical instrument control." *Adaptive and Learning Systems* 1706: 244–255.

[7] McMillen, K. A. 2008. "Stage-worthy sensor bows for stringed instruments." In *Proc. Intl. Conf. on New Interfaces for Musical Expression (NIME).*

[8] Nagai, M. 2010. *MARtLET*. http://michellenagai.com/Site/MARtLET.html

[9] Rasamimanana, N., Flety, E., and Bevilacqua, F. 2005. "Gesture analysis of violin bow strokes." In *Proceedings of Gesture Workshop 2005 (GW05)*. 145–155.

[10] Trueman, D. 2010. "Clapping Machine Music Variations." In *Proc. Intl. Computer Music Conference (ICMC).*

[11] Wright, M. and Freed, A. 1997. "Open Sound Control: A new protocol for communicating with sound synthesizers." In *Proc. Intl. Computer Music Conference (ICMC).*

[12] Young, D. 2008. "Classification of common violin bowing techniques using gesture data from a playable measurement system." In *Proc. Intl. Conf. on New Interfaces for Musical Expression (NIME).*

MTCF: A framework for designing and coding musical tabletop applications directly in Pure Data

Carles F. Julià
Universitat Pompeu Fabra
138 Roc Boronat
Barcelona, Spain
carles.fernandez@upf.edu

Daniel Gallardo
Universitat Pompeu Fabra
138 Roc Boronat
Barcelona, Spain
daniel.gallardo@upf.edu

Sergi Jordà
Universitat Pompeu Fabra
138 Roc Boronat
Barcelona, Spain
sergi.jorda@upf.edu

ABSTRACT

In the past decade we have seen a growing presence of tabletop systems applied to music, lately with even some products becoming commercially available and being used by professional musicians in concerts. The development of this type of applications requires several demanding technical expertises such as input processing, graphical design, real time sound generation or interaction design, and because of this complexity they are usually developed by a multidisciplinary group.

In this paper we present the Musical Tabletop Coding Framework (MTCF) a framework for designing and coding musical tabletop applications by using the graphical programming language for digital sound processing Pure Data (Pd). With this framework we try to simplify the creation process of such type of interfaces, by removing the need of any programming skills other than those of Pd.

Keywords

Pure Data, tabletop, tangible, framework

1. INTRODUCTION

In the past decade we have seen a proliferation of musical tabletops. Currently, so many "tangible musical tables" are being developed that it becomes difficult to track every new proposal[1].

Independently of the relevant differences that can exist between these systems, scholars tend to agree in the benefits of interacting with large-scale tangible and multi-touch devices. Their vast screens make them excellent candidates for collaborative interaction and shared control [2][4], while favoring at the same time, real-time, multidimensional as well as explorative interaction, which makes them especially suited for both novice and expert users [6]. This last author also states that the visual feedback possibilities of this type of interfaces, makes them ideal for understanding and monitoring complex mechanisms, such as the several simultaneous musical processes that can take place in an interactive digital system for music performance [5].

This growing tabletop popularity, clearly in the musical domain but also in other fields, has increased the publicly available information for the rapid development and prototyping of these types of interfaces. Online communities of DIY builders such as the NUIGroup[2] collect large knowledge bases of resources and many easy-to-follow tutorials are publicly available [12]. The development of this type of hardware solutions has indeed become easier and affordable than ever, allowing practically anyone to experiment with tabletop computing.

From the software side, several well-known open-source solutions do also exist, both for the tracking of multi-touch fingers, such as the NUIGroup's Community Core Vision[3], or for the combined tracking of fingers and objects tagged with fiducial markers, such as reacTIVision [1]. These and other existing software tools greatly simplify the programming of the input component, essential for this type of interfaces, but this solves only one part of the problem. The visual feedback or the graphical user interfaces, which do often also include problems specific to tabletop computing, such as aligning the projector output with the camera input or correcting the distortion that results from the use of mirrors, still have to be manually programmed. Not to mention the underlying musical engine, our main reason after all for developing this type of applications.

Taking into account these considerations, it may be difficult to acquire the required skills for being capable of programming the visual interface and the audio component, even to find a single programming language or framework supporting well these two components.

A simple solution to this last problem, as presented in previous papers such as [4][3], is to divide the project into two different applications: one focused on the visual feedback and another focused on the audio and music processing. However, dividing the tasks will not eliminate the need for programming still on both sides. The system we present here has been designed for simplifying these technical difficulties.

2. MUSICAL TABLETOP CODING FRAMEWORK

Musical Tabletop Coding Framework(MTCF) is an open source framework for the creation of musical tabletop applications that takes a step forward in simplifying the creation of tangible tabletop musical and audio applications, by allowing developers to focus mainly on the audio and music programming and on designing the interaction at a conceptual level (because all the interface implementation will be done automatically).

MTCF provides a standalone program for the visual in-

[1]Kaltenbrunner, has a website devoted to Tangible Music, which includes a quite exhaustive list of devices: `http://modin.yuri.at/tangibles/`

[2]`http://nuigroup.com`

[3]`http://ccv.nuigroup.com/`

NIME'11, 30 May–1 June 2011, Oslo, Norway.

terface and the gesture recognition, which communicates directly with Pd[11], and which enables programmers to define the objects and their control parameters, as well as the potential relations and interactions between different objects, by simply instantiating a series of Pd abstractions. MTCF can be freely downloaded at github[4].

2.1 Description of the system

MTCF has been designed for being used in conjunction with any type of tabletop surface that supports both the detection of marked tangible objects and multitouch interaction, although it does not force both interaction modes. The only restriction for the hardware is the output protocol used, its tracking system should complie with the TUIO protocol [9]. Otherwise it does not impose either any restriction on the size or shape of the surface, allowing to design for rectangular surfaces as well as for circular ones such as the Reactable.

Our internal test hardware is the one used for the reactable [7] and reacTIVision[8] as the tracking software (see Fig. 1). The generated data from reacTIVision (i.e. position and orientation of all the tagged pucks and fingers) is sent to MTCF using the TUIO protocol. MTCF just monitors all the incoming TUIO messages and sends them filtered to Pd by means of the Open Sound Control (OSC) protocol[13]. From Pd, control messages and waveform data are also transmitted back to MTCF, that is in charge of permanently refreshing the visual display.

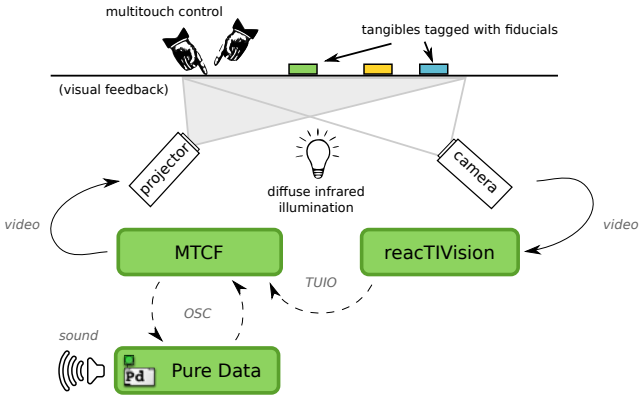


Figure 1: System diagram.

2.2 MTCF: Dealing with the Input data and with the GUI

MTCF is itself implemented on top of openFrameworks[5] (OF), a group of multi-platform libraries written in C++, specially designed for assisting creative applications programming.

MTCF also uses an external OF add-on, ofxTableGestures, which we had previously implemented with the aim of assisting multi-purpose (i.e. not necessarily musical) tabletop application design. ofxTableGestures does already solve some of the typical problems that appear in the development of generic tabletop applications, such as dealing with the tracking incoming messages or correcting the graphical output distortion or alignment. But ofxTableGestures is meant for OF programmers, which means that for using it, programming in C++ is still needed. In that sense, MTCF, built on its turn on top of ofxTableGestures, can be seen as a specialised and simplified subset of ofxTableGestures: while it does not permit all of ofxTableGestures' functionalities, it simplifies enormously the programming tasks by putting everything on the Pd side. Although no understanding of how ofxTableGestures works is needed for fully exploiting MTCF potential, next we will describe some of the basic ofxTableGestures features in order to give a clearer idea of the whole architecture.

ofxTableGestures is itself divided in two parts: TUIO input and graphics output. ofxTableGestures' TUIO input part processes the messages that arrive to the framework from any TUIO-compliant application (e.g. reacTIVision). Once these messages are processed, this component detects and generates gestural events for the top-level programmer. ofxTableGestures's graphics part on its side, helps to create drawable objects while applying the distortion correction to everything that is drawn. ofxTableGestures also includes a self-contained tabletop simulator, which simulates figures and multiple fingers interaction, allowing testing the applications without the need of a real table. (see Fig. 2). When the simulator is enabled, a right panel with a subset of figures is shown on one side of the screen. These figures are labelled with the identifier that will be reported by YUIO messages to the system. In order to maintain the fidelity between the physical table and the simulator, the figures used on the simulator match in size and shape with the real ones in our setup. ofxTableGestures includes six different figure shapes (circle, square, star, rounded square, pentagon and dodecahedron), which are defined in a configuration file that includes the figure shape, the figure identifier and the figure colour.

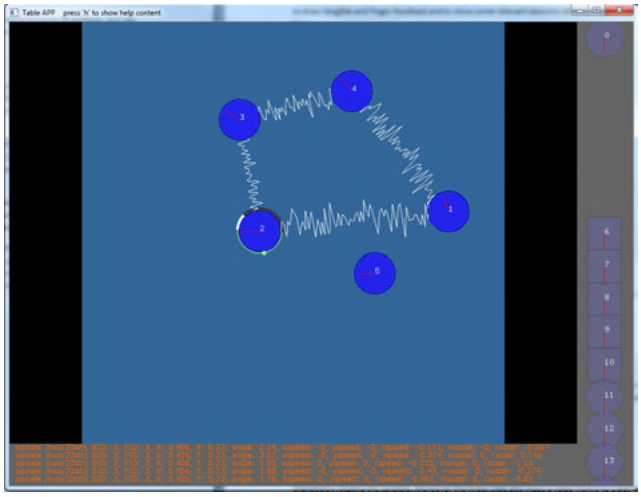


Figure 2: Simulator screen shoot.

MTCF receives data from the TUIO application, processes it, displays the graphic feedback and sends the filtered data to Pd via OSCMessages. At this stage, MTCF only draws the figure shapes and the fingers' visual feedback, all in their correct positions. The remaining graphics (such as the waveforms and the relations between the figures) are drawn in a second step, according to the additional information that is send back via OSC messages from Pd to MTCF. This will be addressed in the next section.

By default, MTCF pucks only convey three basic parameters: X position, Y position and rotary angle. Additional parameters can be enabled from Pd for any specific object. This optional additional information includes parameters resulting from the relations between pairs of pucks (distance and angle between them) as well as parameters resulting from the finger interaction onto given pucks, which

[4] https://github.com/chaosct/Musical-Tabletop-Coding-Framework/downloads
[5] http://www.openframeworks.cc/

can have two extra widgets (Object bar and finger slider) that can be activated from Pd, as shown in Fig. 3. These parameters are displayed as two semicircular lines surrounding the puck, keeping the orientation towards the centre of the table.

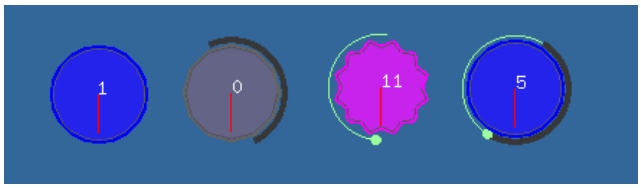


Figure 3: Tangibles with different feedbacks and controllers.

Objects' bars convey a value between 0 and 1 that can be changed by rotating the tangible. The finger slider, represented by a thinner line with a dot that can be moved using a finger, also ranges between 0 and 1. In the next section we will concentrate on the Pd side of MTCF.

2.3 Using MTCF from Pure Data

MTCF was designed to be used along with Pd, as this has become one of the most popular languages for realtime audio processing and programming. The main idea of this framework was to allow expert Pd users to interface their patches using a tangible tabletop setup. For this, MTCF provides nine Pd abstractions that transparently communicate with MTCF, and that are used to define the objects, the relations between them, and the data the programmer wants to capture from the tabletop interface. Not all of these abstractions have to be always used, as this will depend on the affordances of our musical application interface.

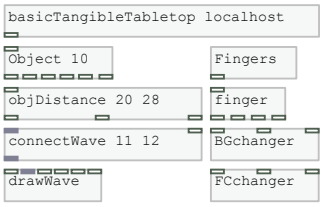


Figure 4: MTCF Pd Abstractions.

Only one abstraction is mandatory and responsible for all OSC communication between the Pd patch and MTCF: `[basicTangibleTabletop]`. Its single argument is the address of the computer running MTCF. This will typically be *localhost*, although changing this address can be useful in some situations, such as in testing several projects (on different laptops) wit only one tabletop (only running the visual part). One and only one instance of this object must exist in the Pd program.

2.3.1 Defining Objects and Parameters

Some additional abstractions will allow us to define what physical pucks will be used on the application. Instantiating `[Object` n`]` will tell the system to include the object with the id code n.

As described in the previous section, a slider plus a $[0, 1]$ rotatory parameter can be activated around any puck. The (de)activation of these extra controllers is done in Pd, by sending messages to their associated `[Object]`. Only when these elements are active Pd will receive this additional information.

Outlets in `[Object]` output the presence of the puck (Boolean), its position, orientation, and if activated, its slider and rotary parameter values.

Inspired by the Reactable paradigm, which allows the creation of audio processing chains by connecting different objects (such as generators and filters), MTCF also permits to use the relations between different pucks and can make them explicit. However, unlike the Reactable, MTCF is not limited to the creation of modular, subtractive synthesis processing chains; any object can relate to any other object independently of their nature. This allows for example to easily create and fully control a tangible Frequency Modulation synthesiser, by assigning each carrier or each modulator oscillator to a different physical object; or a Karplus-Strong plucked string synthesiserby controlling the extremes of a virtual string with two separate physical objects.

On the counterpart, MTCF does not yet permit dynamic patching [10], so it is not capable of producing a fully functional Reactable clone, neither was this its main objective. In MTCF, the connections between the pucks have to be made explicitly by the programmer in the Pd programming phase. This is attained by using `[objDistance` m n`]`, which continuously updates about the status of this connection, and (if existent) about the angle and distance between objects m and n. The programmer can also specify whether she wants this distance parameter to be drawn on the table by sending a Boolean value into the `[objDistance]` inlet.

2.3.2 Drawing Waves

Also inspired by the Reactable, MTCF can easily show the "sound waves" going from one object to another. This can be achieved by using the `[connectWave]` object. This abstraction takes two parameters that indicate the two object numbers between which the wave should be drawn. As indicated before, this waveform does not necessarily indicate the sound coming from one object into the other, but can rather represent the sound resulting from the interaction between two combined objects, or from any other sound thread from the Pd patch.

An audio inlet and an outlet are used to take the waveform and to act as a gate, allowing the audio to pass, only if the two pucks (and therefore the waveform) are on the surface. This ensures that no unintended sound will be processed neither shown when its control objects are removed. Additionally, a control inlet lets the patch to activate and deactivate this connection.

This way of drawing waveforms has some consequences: first, waveforms are drawn by default between pucks, difficulting the drawing of waveforms between two arbitrary points, or from one object to the centre, as Reactable does. This can be overcome by using a simpler Pd abstraction, `[drawWave]`, which has exactly this very purpose: drawing waves between two points.

The second but very important consequence is that the audio connection between two physical pucks is a Pd object itself. Instead of having Pd audio connections between `[Object]` abstractions, the programmer must therefore use `[connectWave]` abstractions, which simply send the waveform information to MTCF for drawing it. This can be confusing, specially when chaining multiple physical pucks imitating an audio processing chain, since the programmer must then consider all the possible combinations (Fig. 5).

2.3.3 Extra features

For more advanced interaction, additional abstractions are also provided. `[Fingers]` gives full information of the position of all fingers detected on the table, while `[finger]` can be used to extract individual fingers information (see Fig. 6). These abstractions can be used to control less obvious parameters.

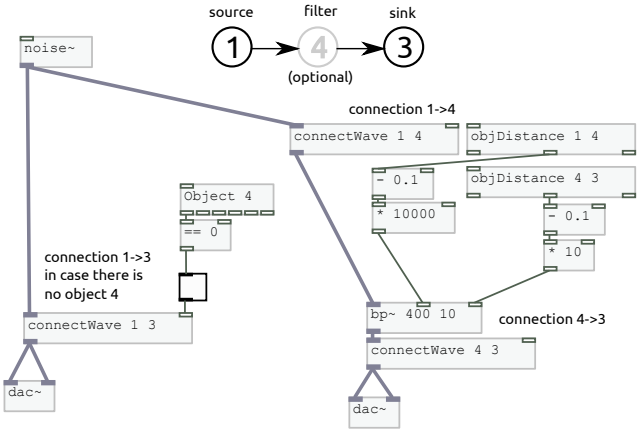


Figure 5: A processing chain example. Puck 1 is a noise generator, puck 4 is a filter, and puck 3 is an audio sink (i.e. the speakers). The programmer must consider the connections both when puck 4 is present ($1 \rightarrow 4 \rightarrow 3$) and when it is not ($1 \rightarrow 3$).

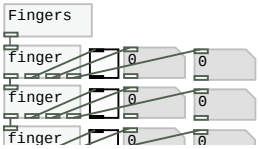


Figure 6: A Pd structure to receive information of the several fingers on the surface.

Two additional abstractions can be used for visual purposes: `[BGchanger]` and `[FCchanger]` respectively allow changing the background colour of the tabletop and the colour of the fingers' trailing shadows. Changing colours, for example according to audio features, can create very compelling effects.

3. CONCLUSIONS

The experience we have gained until now from using MTCF on two short half-day workshops, indicate that MTCF is not only a very valuable tool for the quick development and prototyping of musical tabletop applications, but also an interesting system for empowering discussion and brainstorming over some concepts of software synthesis control and interaction.

We are also aware that there are many issues that can still be improved. While Pd experts quickly understand the framework's mechanisms and take full profit from it producing interesting results in very short times, a few advanced users missed some higher level control possibilities. At its current stage, MTCF is clearly very oriented towards real-time sound synthesis and processing control, lacking of higher level and more structural controls that could communicate with Pd entities such as data arrays or sequences. In a near future, we therefore plan to include more graphical interface features, probably making a more extensive use of multi-touch interaction, in order to be able to control time-oriented and structured data such as envelopes or sequences of events.

4. ACKNOWLEDGMENTS

This work has been partially supported by TEC2010-11599-E (Ministerio de Ciencia e Innovación, Gobierno de España) and by Microsoft Research Cambridge.

5. REFERENCES

[1] R. Bencina, M. Kaltenbrunner, and S. Jordà. Improved topological fiducial tracking in the reactivision system. In *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, page 99. Ieee, 2005.

[2] Y. Fernaeus, J. Tholander, and M. Jonsson. Beyond representations: towards an action-centric perspective on tangible interaction. *International Journal of Arts and Technology*, 1(3):249–267, 2008.

[3] L. Fyfe, S. Lynch, C. Hull, and S. Carpendale. Surfacemusic: Mapping virtual touch-based instruments to physical models. In *Proceedings of the 2010 conference on New interfaces for musical expression*, pages 360–363. Sydney, Australia, June 2010.

[4] J. Hochenbaum, O. Vallis, D. Diakopulos, J. Murphy, and A. Kapuy. Designing expressive musical interfaces for tabletop surfaces. In *Proceedings of the 2010 conference on New interfaces for musical expression*, pages 315–318. Sydney, Australia, June 2010.

[5] S. Jordà. Sonigraphical instruments: from fmol to the reactable. In *Proceedings of the 2003 conference on New interfaces for musical expression*, NIME '03, pages 70–76, Singapore, Singapore, 2003. National University of Singapore.

[6] S. Jordà. On stage: the reactable and other musical tangibles go real. *International Journal of Arts and Technology*, 1:268–287, 2008.

[7] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international Conference on Tangible and Embedded interaction*, pages 139–146. ACM, 2007.

[8] M. Kaltenbrunner and R. Bencina. reacTIVision: a computer-vision framework for table-based tangible interaction. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 69–74. ACM, 2007.

[9] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. Tuio-a protocol for table based tangible user interfaces. In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005), Vannes, France*, 2005.

[10] M. Kaltenbrunner, G. Geiger, and S. Jordà. Dynamic patches for live musical performance. In *Proceedings of the 2004 conference on New interfaces for musical expression*, NIME '04, pages 19–22, Singapore, Singapore, 2004. National University of Singapore.

[11] M. Puckette. Pure Data: another integrated computer music environment. *Proceedings of the Second Intercollege Computer Music Concerts*, pages 37–41, 1996.

[12] J. Schöning, P. Brandl, F. Daiber, F. Echtler, O. Hilliges, J. Hook, M. Löchtefeld, N. Motamedi, L. Muller, P. Olivier, et al. Multi-touch surfaces: A technical guide. *Technical Reports of the Technical University of Munich*, 2008.

[13] M. Wright and A. Freed. Open sound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the 1997 International Computer Music Conference*, pages 101–104, 1997.

Physical modelling enabling enaction: an example

David Pirrò
University of Music and Performing Arts
Institute of Electronic Music and Acoustics
Graz, Austria
pirro@iem.at

Gerhard Eckel
University of Music and Performing Arts
Institute of Electronic Music and Acoustics
Graz, Austria
eckel@iem.at

ABSTRACT

In this paper we present research which can be placed in the context of performance-oriented computer music. Our research aims at finding new strategies for the realization of enactive interfaces for performers. We present an approach developed in experimental processes and we clarify it by introducing a concrete example. Our method involves physical modelling as an intermediate layer between bodily movement and sound synthesis.

The historical and technological context in which this research takes place is outlined. We describe our approach and the hypotheses on which our investigations ground. The technological frame in which our research took place is briefly described. The piece *cornerghostaxis#1* is presented as an example of this approach. The observations made during the rehearsals and the performance of this piece are outlined. Grounding on ours and the performers' experiences, we indicate the most valuable qualities of this approach, sketch the direction our future experimentation and development will take, pointing out the issues we will concentrate on.

Keywords

Interaction, Physical Modelling, Motion Tracking, Embodiment, Enactive interfaces

1. INTRODUCTION

Starting from the first and still fundamental attempts to translate body movements into sound by Theremin (the Theremin, the Terpsitone), the design of interfaces for interaction has been a key issue not only in technology development but also for the theoretical discourse around computer music. Questioning and exploring the role of the performer and the performance [2, 8] and the possibilities of the integration of these into electronic and computer music has been since then a central matter of discussion.

In recent years the availability of faster computers allowing real-time sound processing and motion tracking, opened new possibilities for interaction design and gave a great impulse to research. A multitude of novel mapping strategies were developed striving not only to cope with this newly available possibilities but also to find meaningful ways to couple movement and sound. The search for connections of sound and music to movement and gesture has been approached from an aesthetic research standpoint [5] and the embodiment and enaction discourse [1, 12] offered new viewpoints to the development of musical interfaces. From a poietic perspective, possibilities to track movement have been extensively used and questioned in many artistic projects (i.e. Rokeby's VNS[1] or the SICIB system[9]) in particular involving the integration of dance (i.e. the DIEM project[2]) aiming at achieving a high degree of embodiment in sound and music production (the EGM project [3]).

Research and development in this field is not only a scientific and technological challenge, but also an artistic and musical necessity. Interaction design has become a central compositional issue. On the one hand, composers long for strategies and techniques that allow them to "compose" their instruments [10] and interfaces. On the other hand, the performers need to be enabled to enact the sound and music generation through the interfaces they are presented with rather than to merely control them.

Searching for new possibilities in these respects, we propose here an approach of interaction design (in particular using motion tracking) that uses physical modelling as an intermediate layer between the performers' actions and the sound synthesis (including its control).

2. THE APPROACH

Our aim is to provide performers with an interaction system, an environment that they can intuitively learn and cope with. In order to achieve these qualities we search for a method allowing to address the players' tacit bodily knowledge. The strategy we developed relies on the design and implementation of physical models.

Physical modelling is a well known technique in sound synthesis. One of its strengths lies in an intuitive control of the sound synthesis. This method has been widely used in conjunction with various motion tracking technologies to interactively produce sound [6, 11].

In the approach presented here, physical modelling is not used as a sound synthesis engine. Rather it is an intermediate layer placed between the input from the performers' actions and the sound processing or synthesis layer (c.f. figure 1). The intermediate physical modelling layer constitutes an interface at the point where body movement and sound generation or composition meet. The tracked performer interacts with the physical model causing a change in its state. These changes are then used to control the sound synthesis engine. Eventually the resulting sound will reflect the reactions of the physical model and will provide the feedback for the user.

[1]D. Rokeby. Very nervous system, http://homepage.mac.com/davidrokeby/vns.html
[2]The Royal Academy of Music. Aarhus - diem, http://www.musikkons.dk/index.php?id=300

NIME'11, 30 May–1 June 2011, Oslo, Norway.

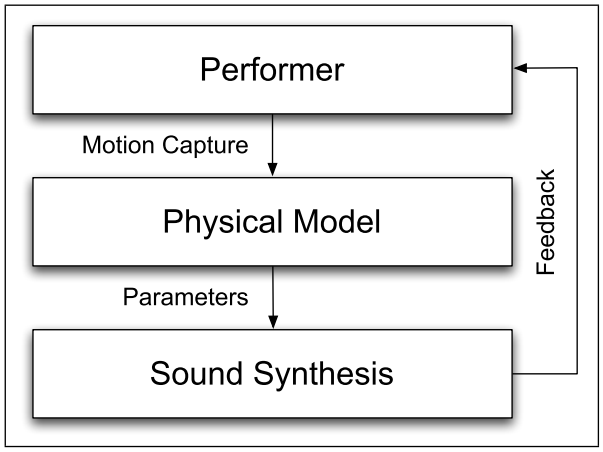


Figure 1: Diagram illustrating our approach to interaction design.

Our fundamental hypothesis is that the behaviour the performers are confronted with by interacting with the physical model, belongs to their intuitive knowledge of the physical world. The reaction generated by the simulation in response to the performers' actions is induced in the sound and exhibits dynamics and behaviour familiar to the performers as it resembles the qualities of our interaction with the real physical world. This interface, by tapping into embodied knowledge and activating already acquired motor skills is an enactive interface - an interface through which it should be possible to truly enact the sound generation and the composition.

The physical model ensures a coherence and continuity in the sound output that is, in our experience, most important for the performers. They can rely on it, can engage with it and possibly play with it as an instrument. Further there is a certain degree of "sensory predictability" inherent to such models. This predictability is not to be understood in a strictly mathematical sense, as even very simple models can be very difficult to predict. Rather this term is used here to indicate the felt consistency of the effects with the performed actions that allows for instinctive guesses about which effects can be expected.

In the setup we delineated (figure 1) the sonic feedback plays an important role. The sound alterations resulting from the interaction with the model are the only available feedback. As such the sound response has to be designed in a way that its changes are easy to follow and to relate to the physical properties of the model. Later we will present a piece, *cornerghostaxis#1*, in which physical modelling was applied in the sense we describe here. In this piece, spatialization was used as the primary cue for the performer to follow the changes in the model and the movements of the modelled objects present in the virtual scene. However there are surely other types of sound manipulation that can carry the information coming from the model in a clear way. The approach we propose and the software we developed so far is open for such possibilities.

We believe that this approach not only provides the performer with an enactive interface but also offers the composer, sound artist or interaction design researcher an intuitive way to conceive the interaction, realize it, and refine it.

3. SETUP

In this section we briefly describe the software and hardware environment we worked with while developing, rehearsing and performing *cornerghostaxis#1*, the piece we will describe in the next section.

3.1 Tracking

The IEM CUBE is the research environment in which our experiments were carried out and the piece was rehearsed. Physically it is a 120 m2 studio space equipped with a 24-channel hemispherical Ambisonics-based sound projection system, which is complemented by an array of 48 ceiling-mounted speakers. Besides the sound projection and rendering infrastructure, a VICON motion-capture system with 15 infrared cameras is installed allowing for high-quality rigid body or full-body motion tracking. A tracking rate of 120 fps is used at which the position and orientation data is provided by the system. At this frame rate the system resolves positions in 3D-space with a precision of about 1 mm.

3.2 Software

For the design of the physically modelled scene used in the piece we developed a software framework in the SuperCollider language. This framework allows rapid prototyping of the physical models, manages motion tracking input and provides a simple 3D visualization. The software has been designed with the openness and flexibility in mind required to react to the particular needs of a project or composition using our interaction model. The new SuperCollider extensions comprise a set of tools for managing and conditioning of motion tracking data arriving via OSC and greatly simplify the process of designing different virtual spaces or "scenes" in which the objects (masses) subject to the physical modelling are added, placed or removed.

The forces acting between the objects can be freely defined starting from a set of the predefined forces to choose from (spring, gravitation, electrostatic force, etc.). Most important is the possibility to define particular constraints that restrict the motion of the objects in different ways.

Aiming at establishing a connection between the virtual space of the physical model and the performing space, the possibility to define the positions of loudspeakers in the virtual space was introduced. Further, using the distances to these virtual loudspeakers, a DBAP (Distance-Based Amplitude Panning[7]) algorithm is used for the spatialization of the sounds "carried" or modified by these objects.

4. CORNERGHOSTAXIS#1

In this section we introduce and illustrate *cornerghostaxis#1*, an artistic work that was developed using the strategy we described earlier. Through this concrete example we hope to provide a better understanding of how we intend physical modelling to be applied in interaction design.

cornerghostaxis#1 was premiered February 27th 2009 at the Cube of IEM Graz, during IMPULS Academy 2009 in the context of the Motion-Enabled Live-Electronics workshop [4] (bassoonist: Dana Jessen) and was also invited to the Bodily Expression in Electronic Music Symposium (BEEM) at Mumuth Graz and performed on November 7th 2009 (bassoonist: Stephanie Hupperich).

4.1 The piece

cornerghostaxis#1 is an electroacoustic composition for solo bassoon. The piece is the result of the collaborative effort of a three people team: Stephanie Hupperich (bassoon), Gerriet K. Sharma (composition) and David Pirrò (physical modelling / interaction design). The aim was to design an environment in which the player interacting with the physical model establishes a gestural and bodily relationship between the sounds she plays on her instrument, her

Figure 2: Stephanie Hupperich performing *Cornerghostaxis#1* at the MUMUTH in Graz on November 7th 2009. The tracking target, made up of five infrared reflecting spheres, is attached to the instrument (upper left corner).

movements in space and four electronic sources that are dynamically spatialized on a loudspeaker array.

In the piece the position and orientation of the tracked instrument is used as input for a physical model. The virtual space in which the physical simulation is taking place is a representation of the real space in which the performance took place including the positions of the loudspeakers and the instrument. The physical objects that move and interact in this space are constrained on the surface of a hemisphere on which also the loudspeakers are placed, reflecting their actual positions. The involved objects have a very clear relationship: one can imagine them as electrically charged masses with the same charge. Thus the forces acting between the objects are repulsive.[3]

The tracking data is used to control the position and orientation of a square with four "charged" masses placed at its corners. The other masses are free to move on the hemisphere spanned by the loudspeakers: they are also "charged" and repelled by the previous ones as well as from one another. The distances of these masses to the virtual loudspeakers are used to control a DBAP algorithm that determines how the four channels of the tape composition by Gerriet K. Sharma are spatialized on the physical loudspeaker array. Furthermore, the amplitude of the four sources is controlled according to the movement speed of these masses and depending on the distance to performer. If the performer is close to one of them (i.e. she "captured" one, see below) that source grows louder.[4]

The piece has been always conceived as a whole, and the development of each parts advanced in parallel to the others. The physical model is not just an effect used to spatialize the tape composition, but it is part of the piece, part of the environment in which the composition unfolds.

In the next section we try to summarize how the approach described before in section 2 reshaped the working routine in the explorations and rehearsals of the piece, with respect to our aims. We therefore collect the most important observations being made by the performers and by us. But we also attempt to condense our reflections based on our own aesthetic experiences gathered throughout the process leading to the realization of the piece.

We understand the whole realization of the piece, beginning with the design of the physical model, passing on to the preliminary explorations with the performer, to the rehearsals and the final performance, as part of an experimentation aimed at putting into practice the strategy we described and observe what and how it "happens". An interpretation or evaluation of these observations is not explicitly given, but will be the object of future research.

4.2 Observations

The most important feedback was given to us by the performers. The musicians involved in the development of the piece underlined that they felt having achieved a clear understanding of the dynamics of the sound spatialization and how they could influence it.

They could quickly established an intimate control of the interface and they could rapidly learn how to play it.

This understanding also changed the communication between musician, composer and programmer. Relying on the physical metaphor, on which the programming and the whole realization of the piece are basing, the performers could more easily communicate with the composer and programmer. In this sense the intermediate physical modelling layer appears as a platform for the exchange and refinement of ideas which are shared among all the participants, regardless of their technical knowledge. For example asking "Could you make the masses *heavier*?" is straightforward for the performer. At the same time it is easy for the programmer to understand and, knowing the model, to accomplish. This is one of the main reasons the performers were actively involved in the setting-up and the development of the piece.

Basically, in performing the piece the musician and the masses play a "hide-and-seek" game. The sources try to escape the performer, always placing themselves at the points most distant to her. This dynamic became very quickly clear to the performer in the first experimental session and her instinctive reaction was trying to find ways of stopping their continuous slipping, blocking one of them by pinning it down, "capturing" it. Also during the performance, the aim for the performer is to "catch" one precise mass out of the four, at a specific moment of the score. But the sound sources, which represent the mass positions in the model, seem to have their own will and try to hinder the musician to achieve her goal, to "win" the game.

It is important to note here that understanding the rules of the play means to understand the laws on which the physical model is based, which are coherently and continuously followed by the simulation and which are inscribed in the sounds' positions and movements. In our experience this gaming quality greatly contributes in making the interaction more clear, interesting and engaging.

The reactions of the model are complex but retain a certain predictability (in the sense already explained in section 2). Thus the performer does not have the perception of erratic reactions of the model, which would destroy the illusion of a coherent environment. However the model and the sources are very difficult to control. It is tough to achieve exactly what the composer or the performer wants. The model "resists" at any moment to the performer's actions, at the same time offering a great detail in interaction, as every little position or rotation changes have audible consequences. In our observations the resistance of the model coupled with the refinement of control, greatly enhances the embodiment. As a matter of fact, the musicians, after a short time of experimenting with the model, feeling challenged, asked for for a more difficult setup, which was

[3]A short video of the model's simulation is available at http://pirro.mur.at/nime11/CGA-Model.mov

[4]A documentation video of the performance at Mumuth Graz is available at http://pirro.mur.at/nime11/CGA.mp4

initially kept simple. That meant more resistance of the environment to their actions, but also more detail for their control.

Resistance and detail of control create a continuous tension between performer and model that can be seen and felt clearly. This tension captures attention and causes engagement for the musician as well as for the audience assisting at the performance.

Given the features of the interaction we described, the performer could fully engage in the play with the environment and with the piece itself. The consistency of the interaction qualities and the resulting sonic feedback, caused a "suspension of disbelief" for the performer, who could truly and bodily trust the coherence of the model's responses, of the connection between her movements and the reactions of the sources. This link was so clear to one bassoonist that she started giving them a "body", regarding them (in her own words) as "colleagues", like she would do with other human players. Furthermore she reported an enhanced sensibility not only in the perception of the spatial location of sound, but also of her own movements, her position in space as well as an increase of her proprioception.

We underline at this point that the model was neither visible to the audience nor to the performer, neither during the rehearsals nor the concerts. It was not clear to the viewer how the model works or exactly which forces were acting in the simulation, as this was not explained before the concerts. It was not our aim to make this aspect evident. In our approach the intermediate physical modelling layer is not intended to be clearly perceivable as such, but its purpose is to enhance the enactivity of the interaction, creating the qualities we described.

Nonetheless, during the informal discussions that took place after the performances, it appeared that the relationship between movement and sound, between action and spatialization, between the player's sounds and the electronic sounds was clear also to the audience attending the performances. The player's efforts, engraved in the qualities of her playing as well as in her body could be seen and could be conveyed to the spectator.

5. CONCLUSIONS

Using physical models as intermediate interaction layer, at least in the example reported, proved to be a very fruitful approach towards the design of enactive environments. This method clearly enhanced the qualities we are trying to achieve, bringing them to light as well as exposing new issues to our observation, which will be addressed in future research.

We think that one of the most interesting features of this strategy is that the performer could play with the electronics as she would in a game. This aspect appears to be of central importance for the "suspension of disbelief" experienced by the performer. The rules and the aim are clear for her and for her counterparts (the sources) and as long as the game unfolds coherently and the reactions remain in a range of predictability, the musician is more interested in playing (and winning) the game than asking herself how she should relate to the electronics and how things work on a technical level.

The physical model resists to the players actions. The modelled objects try to impose their dynamics and behaviour, but at the same time offer to the performer a great variety of ways to guide and control them. These open a whole space of possibilities which is tightly connected to the resulting effects in the model: for example fast and big movements have different effects than slow and little movements. The reactions of the sources scale accordingly to the spatial and temporal qualities of the performer's efforts in opposing their intentions to the environment. This results in a very complex, detailed and rich interactivity and appears as a central quality of the approach we describe.

As we described in section 2, in our approach we employ physical modelling to design an interface that, by tapping into the performers' own embodied knowledge, can be regarded as an enactive interface. In the course of our experiments we realized that an important aspect is that by taking this quality of the interface for granted, it is possible to focus on the connection between physical model and the control of sound synthesis and of the composition and how it is realized. This link (figure 1) will be the central object of further research. Until now we used spatialization as primary feedback from the simulation for the musician. Making this connection available to composition would mean to provide the performer with new possibilities to interact with the composition on different levels. This aspect is strongly related to the type of sonic feedback the performers receive and how this would reflect their interaction with the model and with the composition.

6. REFERENCES

[1] N. Armstrong. *An Enactive Approach to Digital Musical Instrument Design*. Phd thesis, Princeton University, 2006.

[2] J. Chadabe. Interactive composing: An overview. *Computer Music Journal*, 8(1):22–27, 1984.

[3] G. Eckel and D. Pirrò. On artistic research in the context of the project embodied generative music. *Proceedings of the 2009 International Computer Music Conference*, pages 541–544, 2009.

[4] G. Eckel, D. Pirrò, and G. K. Sharma. Motion-enabled live electronics. *Proceedings of the 6th Sound and Music Computing Conference*, pages 36–41, 2009.

[5] R. I. Godoy. Motor-mimetic music cognition. *Leonardo*, 36(4):317–319, August 2003.

[6] D. M. Howard and S. Rimell. Real-time gesture-controlled physical modelling music synthesis with tactile feedback. *EURASIP Journal on Applied Signal Processing*, 2004(1):1001–1006, 2004.

[7] T. Lossius, P. Baltazar, and T. de la Hogue. Dbap - distance-based amplitude panning. *Proceedings of the 2009 International Computer Music Conference*, 2009.

[8] F. R. Moore. The dysfunctions of midi. *Computer Music Journal*, 12(1):19–28, 1988.

[9] R. Morales-Manzanares, E. F. Morales, R. Dannenberg, and J. Berger. Sicib: An interactive music composition system using body movements. *Computer Music Journal*, 25(2):25–36, 2001.

[10] N. Schnell and M. Battier. Introducing composed instruments: Technical and musicological implications. *NIME '02 Proceedings of the 2002 conference on New interfaces for musical expression*, pages 138–142, 2002.

[11] M. M. Wanderley and P. Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE 2004*, 92(4):632 – 644, April 2004.

[12] D. Wessel. *An Enactive Approach to Computer Music Performance*. Studio Gramme, Lyon, France, 2006.

SoundGrasp: A Gestural Interface for the Performance of Live Music

Thomas Mitchell
University of West England
UK
tom.mitchell@uwe.ac.uk

Imogen Heap
Megaphonic Records Ltd.
UK
info@imogenheap.com

ABSTRACT

This paper documents the first developmental phase of an interface that enables the performance of live music using gestures and body movements. The work included focuses on the first step of this project: the composition and performance of live music using hand gestures captured using a single data glove. The paper provides a background to the field, the aim of the project and a technical description of the work completed so far. This includes the development of a robust posture vocabulary, an artificial neural network-based posture identification process and a state-based system to map identified postures onto a set of performance processes. The paper is closed with qualitative usage observations and a projection of future plans.

Keywords

Music Controller, Gestural Music, Data Glove, Neural Network, Live Music Composition, Looping, Imogen Heap

1. INTRODUCTION

This work began with a discussion between the authors of this paper regarding intuitive methods by which live musical performance processes can be controlled by simple gestures. The intention was to enable a performer to manipulate digital musical processes without having to defer audience engagement to undertake subtle interactions with machinery.

Since the earliest discussions and observations of computer-based electronic music performances, a recurring theme is the breakdown between the actions of the performer and the effect that these actions have on the sound which is produced. That is, the *transparency* of the mapping between the input to an instrument/device and its corresponding output [5]. Unlike traditional acoustic instruments, the control mapping for modern electronic music devices is often opaque and thus difficult for audiences to infer. Bahn *et al.* [2] argue that traditional notions of musicianship should be maintained in electronic music and consequently the connection between gesture and sound should be preserved. However, other authors contend that phlegmatic performances emanating from the glow of a laptop screen mark an inevitable evolution in contemporary, computer-mediated performance [15]; a change in culture to which audiences must adapt and in many instances already have. In either case, the incorporation of clear sound producing or ancillary gestures into a live performance can enhance both audience engagement [12] and communication between performer and listener [17].

In this work, a live sampler, looper and effects processor are controlled by hand gestures selected to convey the processes that they control. In doing so, the performer is extricated from machine interaction which could be perceived as ambiguous by an audience. The following sections will provide relevant background reading with an overview of the system divided into sections following the strategy proposed in [18] for the development of gestural music devices and mappings. These sections will include:

- the definition of a posture vocabulary,
- the methods by which gestures are acquired and identified,
- the mapping strategy for the assignment of these gestures to the control of audio processes.

2. BACKGROUND

There is a large body of research that examines human computer interaction with hand postures and gestures. A subset of this work is concerned with the use of these techniques for musical purposes. These works can be divided into two broad categories [16]: position tracking methods, using optical, magnetic or acoustic technology; and glove-based methods using electromechanical sensors that directly track fine motor activity. At this stage, SoundGrasp employes a single data glove to sense hand posture, consequently this background section is limited to glove-based input.

2.1 Data Gloves and Music

Since the development of the first data glove in the late 1970s, there have been numerous examples of their use within musical contexts. For example, the Cyber Composer system [10] has been developed to enable the composition and performance of live music using a vocabulary of hand gestures, which are mapped to construct chord and melody sequences. MusicGlove [7] enables a database of multimedia files to be searched and played back using simple hand gestures. Recent examples have seen the mapping of glove-captured gestures for the control of electronic percussion[4] and synthesis [18].

The work presented in this paper focuses on the acquisition of hand gestures and their mapping onto musical processes within a live performance environment. The system enables the realtime sampling and manipulation of sound using gestures that lend themselves to the processes that they control.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

3. LIVE SAMPLING - SOUND GRASPING

Despite the wide musical application of glove-based gestural controllers, live sampling and looping is an area which has been relatively unexplored; although examples are beginning to emerge. One such system is the Vocal Augmentation and Manipulation Prosthesis (VAMP) [11]. Equipped with this device, a singer can 'freeze' a single note when the finger and thumb are pressed together, activating a pressure sensor located on the glove. This 'pinch' gesture captures a short frequency domain representation of the incoming signal which is resynthesised continuously until the pinch is released. Further harmony and amplitude modulation is facilitated through the use of flexion and acceleration sensors also attached to the glove. This mapping ascribes a widely understood gesture for the physical act of 'holding' to a process that 'holds' the incoming audio. Fels *et al.* [5] describe this appropriation of recognised gestures as *metaphor*, which can be used to increase the transparency of control mappings for both audiences and performers.

The second author of this paper regularly performs music incorporating the live sampling of vocals and acoustic instruments. The proposed system has been designed around the requirements of this situation:

1. The musical processes should be controlled without having to defer performativity to engage in machine interaction.
2. There should be a transparent mapping between the input to the gestural controller and the outgoing musical events.
3. Instrumental virtuosity should be compromised as little as possible.

The wearable components of the presented work are shown in Figure 1, comprising a fingerless data glove with a wrist-mounted microphone. This arrangement has minimal constraints on dexterity and unites the gestural controller with the sound capture device. This enables proximal sound sources to be sampled using a grasping metaphor: recording commences when the hand is opened and concludes when the hand is closed. Thus the sound appears to be 'caught' by hand.

Figure 1: SoundGrasp glove with wrist mic

4. SYSTEM OVERVIEW

Gestural music devices are widely represented as a three part system: the gestural controller, the audio processing unit and the mapping that exists between the two [18]. For this work, the mapping and audio processing are both incorporated into a cross-platform C++ application which was developed using the library Juce [14].

Gestural Controller

Figure 1 shows the gestural controller which includes a single 5DT 14 Ultra glove [1] measuring finger flexion and abduction with 14 fibre optic bend sensors. Also connected to the glove is a lavaliere microphone to enable the recording of live input. Both the glove and microphone connect wirelessly to a computer managing the gestural mapping and audio processing.

Gestural Mapping

Raw serial data transmitted by the glove is decoded and routed to the inputs of an artificial neural network to identify discrete and static hand postures. Identified postures are subsequently used to control the state of the audio processing unit.

Audio Processing Unit

The audio processing unit is a software application which currently enables the recording, overdubbing, looping and modification of audio data.

5. POSTURE VOCABULARY

Previous efforts have been made to formalise universal sets of gestures, see for example Henze [8] for gestures associated with media playback. Many of these studies indicate a lack of consensus amongst participants. Consequently, the vocabulary of hand postures adopted for this work has been chosen pragmatically to be identifiably distinct and to enable the use of metaphor in the control mapping. The posture set is shown in Figure 2.

6. GESTURAL MAPPING

The mapping layer of the SoundGrasp system, mediating between the glove and the audio processing unit, consists of three parts: data processing, posture identification and audio control (Figure 3). Data processing involves the unpacking and normalisation of the serial data from the glove into floating-point sensor values in the range 0.0 to 1.0. The details of the posture identification and audio control process are provided below.

6.1 Posture Identification

Posture identification serves to process the calibrated sensor data to identify when the glove has formed a shape approximating a registered posture. This process forms a pattern recognition problem for which artificial neural networks have been demonstrated to be particularly well suited [6].

Artificial Neural Networks

Artificial neural networks provide a biologically inspired machine learning technique which is loosely modelled on the architecture of the brain. The type of neural network employed here is a multilayer perceptron, which is a fully connected feedforward neural network trained with the back-propogation supervised learning technique. This network architecture has been widely used for the non-linear control of audio and visual systems [13]. This section will only

Figure 2: Current posture vocabulary for SoundGrasp

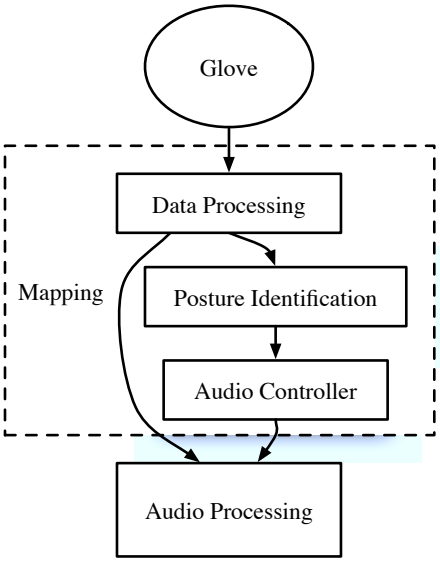


Figure 3: SoundGrasp system architecture

provide a brief summary of the relevant neural network architecture, for fuller treatment and implementation details the reader is referred to [3].

The multilayer perceptron is constructed from layers of interconnected computational units called neurons. Each neuron has one or more inputs and a single output, both of which can be connected either externally or to other neurons. Frequently, the network is configured with three layers: an input layer, a hidden layer and an output layer. The intention is to configure the network such that a known pattern of input values (finger positions) results in a target pattern of output values (identified hand positions). This is achieved with a supervised learning process using sets of training data. A single training set includes a pattern of input and target output values. Subsequent to a successful training procedure, the network should produce a mapping represented by the training set. That is, when the network inputs are set to match an input pattern from the training set, the output of the network should closely match the corresponding training set output pattern.

For the identification of hand postures in this work, the neural network was configured with 14 inputs, matching the quantity of normalised sensor values from the glove. The number of outputs was set to match the number of gestures in the gesture set, currently eight. Subsequent to training the gesture identification was found to be robust with 12 hidden neurons following recommendations set out in [3]. The configuration of the network with one output per posture enabled confidence testing to be performed, preventing the unintentional triggering of postures, while permitting subtle idiosyncrasies that occur when assuming the same hand position.

6.2 Audio Control

Recognised postures are mapped through a further layer, facilitating the selection of audio processes to be controlled using only one glove. This audio control layer manages a simple state based system which enables the performer to switch between modes with sequences of hand postures that form simple gestures [9]. This results in the distinction between two types of gesture:

1. Audio control gestures

2. State/mode control gestures

State control gestures switch the system between different modes which enable the performer to activate different types of audio control processes. This forms a one-to-many mapping between gestures and audio control where a single gesture can be mapped to multiple audio processes through different modes. In establishing the control mapping, audio control gestures, which directly affect the produced sound, use metaphor to increase transparency. In contrast, state control gestures, producing no audible effect, were chosen for performer usability.

Audio Control Processes

The audio control processes were divided into modes which are summarised in Table 1. The principle gesture for audio control is grasping, represented by transitions between postures (g) and (h) in Figure 2. Posture (g) is an open hand, while (h) forms a grasping posture with the tips of each finger in contact with the thumb. Recording is achieved as described earlier and the audio track is cleared with posture (c); raised to the lips, this forms a familiar gesture for silence. In play mode the grasping gesture is reused, playback is paused with (h) and resumed with (g). Reverse playback is initiated with (d) and forwards playback resumed with (g). The filter and effects modes access the sensor data directly with continuous control of the corresponding parameter with the average flexion reading for all four fingers. Lock mode deactivates the glove to enable hand movements without the risk of erroneous audio control, while playing an instrument, for example.

Mode	Audio Processes	Posture
Record	Record/overdub, clear	(h)
Play	Play, stop, reverse	(a)
Filter	Low-pass cutoff	(c)
Reverb	Reveb time	(d)
Delay	Delay time	(e)
Lock	None	(b)

Table 1: Audio Control Processes and Modes

State/Mode Switching

Mode switching is performed with a gesture consisting of two postures in sequence: the first posture (f) indicating the start of a mode switch and the second posture indicating which mode to select. Posture (f) was chosen to initiate the mode switch as full flexion of the lower and upper knuckles of the thumb rarely occurs incidentally. Subsequent mode switch postures are provided in the third column of Table 1.

7. RESULTS FUTURE WORK

Informal testing with a small number of subjects indicated that, after the neural network was trained for individual users, the system was intuitive and easy to lean. Response to hand postures was prompt and stable enabling users to record accurately timed loops consistently. Users were observed to develop their own metaphors adding ancillary gestures over and above those required. For example, several subjects issued audio control gestures with both hands, particularly in the control of playback: amplifying the 'releasing' and 'holding' gestures. The reverse mode, activated with a two fingered point was often accompanied with an additional swipe towards the body, and released with a converse swipe, as if playing an invisible turntable. Some problems were encountered when users wanted to switch modes from postures other than the open hand (g). For example, users wishing to switch modes with playback reversed, record mode disabled or playback halted frequently formed hybrid postures combining the mode switching posture (f) with postures (d) or (h). These issues were solved by adding these hybrid postures to the neural network training set, or by providing the user with further guidance instructions. Alternative solutions will be explored with different neural network architectures to enable thumb postures to be identified in isolation. The authors have many plans for future extensions to this work. Immediate development will incorporate an additional glove and the use of position, orientation and/or acceleration sensors. A second glove hugely increases the degrees of freedom and capacity for further audio and state control switching affording a much more comprehensive range of musical controls. Furthermore, a means of feedback will also be developed as there is currently no mechanism communicating the internal state of the system to the performer. Should a mode switch occur in error, the performer is unaware until the wrong audio processes are subsequently activated. Visual feedback from LEDs attached to the glove will be developed for this purpose.

8. ACKNOWLEGEMENTS

Thanks to Martin Robinson for proof reading this paper and developing the UGen++ library from which the neural network code for this work was ported. For providing the equipment and supporting the project, a big thank you to Professor Tony Pipe and the researchers in the Bristol Robotics Laboratory. Also thank you to the anonymous reviewers of this paper for their suggestions.

9. REFERENCES

[1] Fifth dimension technologies, www.5dt.com, 2011.

[2] C. Bahn, T. Hahn, and D. Trueman. Physicality and feedback: a focus on the body in the performance of electronic music. In *Proceedings of the International Computer Music Conference*, pages 44–51, 2001.

[3] A. Blum. *Neural networks in C++: an object-oriented framework for building connectionist systems*. John Wiley & Sons, Inc., New York, NY, USA, 1992.

[4] S. Chantasuban and S. Thiemjarus. Ubiband: A framework for music composition with bsns. In *Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, 2009.

[5] S. Fels, A. Gadd, and A. Mulder. Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound*, 7(2):109–126, 2002.

[6] S. Fels and G. Hinton. Glove-talk: a neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks*, 4(1):2 – 8, 1993.

[7] K. Hayafuchi and K. Suzuki. Musicglove: A wearable musical controller for massive media library. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2008.

[8] N. Henze, A. Löcken, S. Boll, T. Hesselmann, and M. Pielot. Free-hand gestures for music playback: deriving gestures with a user-centred process. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, 2010.

[9] P. Hong, T. S. Huang, and M. Turk. Gesture modeling and recognition using finite state machines. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[10] H. Ip, K. Law, and B. Kwong. Cyber composer: Hand gesture-driven intelligent music composition and generation. In *Proceedings of the 11th International Multimedia Modelling Conference*, 2005.

[11] E. Jessop. The vocal augmentation and manipulation prosthesis (vamp): A conducting-based gestural controller for vocal performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2009.

[12] G. Paine. Interfacing for dynamic morphology in computer music performance. In *Proceedings of the International Conference on Music Communication Science*, Sydney, 2007.

[13] M. Robinson. Neural networks for audio-visual control. In *Proceedings of the CREAM Symposium, Cybersonica*, 2003.

[14] J. Storer. Juce, www.rawmaterialsoftware.com, 2011.

[15] C. Stuart. The object of performance: Aural performativity in contemporary laptop music. *Contemporary Music Review*, 22(4):59–65, 2003.

[16] D. Sturman and D. Zeltzer. A survey of glove-based input. *Computer Graphics and Applications, IEEE*, 14(1):30 –39, jan 1994.

[17] W. F. Thompson, P. Graham, and F. A. Russo. Seeing music performance: Visual influences on perception and experience. *Semiotica*, 156(1/4):203–227, 2005.

[18] M. Wanderley and P. Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE*, 92(4):632 – 644, April 2004.

Minding the (Transatlantic) Gap: An Internet-Enabled Acoustic Brain-Computer Music Interface

Tim Mullen
Dept. of Cognitive Science and
Swartz Center for Computational
Neuroscience, UC San Diego, USA
tmullen@ucsd.edu

Richard Warp
Composer
1735 Martin Luther King Way
Berkeley, CA, USA
richwarp@gmail.com

Adam Jansch
Dept. of Music
University of Hudderfield
Huddersfield, UK
adam@adamjansch.co.uk

ABSTRACT
The use of non-invasive electroencephalography (EEG) in the experimental arts is not a novel concept. Since 1965, EEG has been used in a large number of, sometimes highly sophisticated, systems for musical and artistic expression. However, since the advent of the synthesizer, most such systems have utilized digital and/or synthesized media in sonifying the EEG signals. There have been relatively few attempts to create interfaces for musical expression that allow one to mechanically manipulate acoustic instruments by modulating one's mental state. Secondly, few such systems afford a distributed performance medium, with data transfer and audience participation occurring over the Internet. The use of acoustic instruments and Internet-enabled communication expands the realm of possibilities for musical expression in Brain-Computer Music Interfaces (BCMI), while also introducing additional challenges. In this paper we report and examine a first demonstration (*Music for Online Performer*) of a novel system for Internet-enabled manipulation of robotic acoustic instruments, with feedback, using a non-invasive EEG-based BCI and low-cost, commercially available robotics hardware.

Keywords
EEG, Brain-Computer Music Interface, Internet, Arduino.

1. INTRODUCTION
Electroencephalography, first applied to humans by Hans Berger in 1924, is the recording of summed electrical activity of large populations of similarly oriented and locally synchronous neurons, located primarily in the human neocortex. Although the earliest effort to sonify EEG was reported in a 1934 paper in *Brain* [1] Alvin Lucier's 1965 *Music for Solo Performer* is widely considered the first EEG-based musical composition. Lucier was strongly motivated by "the image of the immobile if not paralyzed human being who, by merely changing states of visual attention, could communicate with a configuration of electronic equipment" [7]. Interestingly, this was nearly a decade before the earliest published attempts by Jacques Vidal and others to create what we now call a brain-machine/computer interface (BMI/BCI), which is a system that uses signals recorded directly from the brain to manipulate an external actuator [14]. In *Solo Performer* Lucier's amplified alpha (8-12.5 Hz) brainwaves were played through loudspeakers coupled to a battery of percussive instruments, allowing him to generate resonant acoustic events by modulating his alpha rhythm. Lucier's pioneering work was followed by a number of artists and throughout the 1960's and 1970's experimentation with brainwave sonification flourished (see [9] for a review). However, this was followed by over a decade of relative silence.

Within the last decade, due in part to successes in the BCI field, there has been a resurgence of interest in the use of EEG-based BCI technology in musical composition leading Miranda and Brouse to coin the term Brain-Computer Music Interface (BCMI) to refer to systems that use a BCI for musical expression [9,10]. Some BCMI researchers have focused primarily on active control of a musical interface using standard BCI tools; for example, Mick Grierson's adaptation of a P300 speller, which allows a user to construct a sequence of musical notes by attending to various symbols on a display [5]. Others have focused on neurofeedback applications and passive cognitive state detection/sonification [6,15]. Still others have explored collaborative sonification of the mental state of multiple individuals simultaneously. For instance, Steve Mann, James Fung, Ariel Garten and Chris Aimone's *Regen/DECONcert* series had dozens of participants don wearable EEG hardware and alter a synthesized music soundscape via changes in their collective alpha activity [8].

Importantly, most of these and other BCMI systems have incorporated local control of a digital and/or synthesized music interface. There have been comparatively few attempts to create BCMI systems that mechanically control acoustic instruments using EEG. As we shall later discuss, the use of visible, acoustic instrument ensembles, with their somewhat anthropomorphic, unpredictable and thus essentially 'human' method of sound production, introduces new aesthetic opportunities and challenges. Secondly, although a number of artists have explored interactive music creation over the Internet (as reviewed in [10]), comparatively fewer Internet-enabled BCMI installations/performances have been developed. One exception is Andrew Brouse's *InterHarmonium* project [3]. As with any other Internet-enabled interactive media system, including the possibility for distributed communication and interaction in a BCMI may significantly expand the range of possibilities for collaborative musical expression and audience participation.

1.1 Music For Online Performer
On January 16, 2010 we premiered *Music for Online Performer* as part of Adam Jansch and Richard Glover's *In Tones: Organ/Radio/Television/Internet* installation series. The name and other subtle references to Lucier's *Solo Performer* – including the use of acoustic percussive instruments – were

NIME'11, 30 May–1 June 2011, Oslo, Norway.

chosen due to our mutual respect for Lucier's pioneering work. Here electrical signals recorded from the brain of a participant (T.M.) in San Diego, USA were used to manipulate, in near real-time, acoustic instruments in front of a live audience at Phipps Hall at the University of Huddersfield, UK. Using freely-available Livestream™ and Skype™ technology, the music was streamed back to the conductor/composer (R.W.) in San Francisco and the "brainist" in San Diego, who used this feedback (along with local visual feedback), combined with compositional instructions delivered by the conductor, to manipulate his brain rhythms and thereby inform the ongoing composition. In addition, a live Internet audience watched audio-video feeds from all three locations and was in constant communication with the conductor via a Livestream chatroom, allowing them to indirectly influence the composition.

The installation was structured around the concept of a quartet: four instruments being manipulated by four fundamental neuronal frequency bands estimated from four neural signals recorded from the brain of the solitary performer. The installation was also comprised of four participating parties, distributed around the world but connected via the Internet: the brainist, the composer/conductor, the physical audience (Phipps Hall), and the virtual (Internet) audience.

2. TECHNICAL DESIGN

The design schematic for *Online Performer* is outlined in Figure 1. The brainist is seated in a room in front of two displays, a visual neurofeedback display and a compositional instructions display. Stereo auditory feedback is provided via speakers.

2.1 Data Acquisition

64-channel EEG (Biosemi, Inc) is recorded from the brainist at a sampling rate of 256 Hz. The data is imported into Matlab® (Mathworks, Inc) in 2-second segments using the open-source ERICA/Datariver environment [4]. Due to a hardware issue involving Arduino memory buffer maintenance, data controlling the musical instruments could not be updated faster than 5 instructions/sec. Thus, we fixed the time interval between data segments to 200 ms, although this could theoretically be decreased by at least a factor of 10 or more.

2.2 EEG Features

Each 2-second data segment is separated into 64 maximally independent time series (independent components or "ICs") by projection through a spatial filter previously learned on training data by Independent Component Analysis [2,11]. Here, the training data was a 30-minute long continuous EEG time series recorded from the brainist performing a series of mental exercises, similar to those used to control the music BCI (relaxation, left hand motor imagery, right hand motor imagery, mental calculation). Four of these components are selected based on prior analysis of the spatial topography of the components across the scalp. In our implementation, we selected four components each with spatial filter weights resembling the projection of a single equivalent-current dipole (e.g., a patch of locally synchronous neurons constituting an EEG "source") located near one of frontal midline cortex (FMC), visual cortex (VC), or left or right sensorimotor cortex (lSMC, rSMC).

The power spectral density for each selected IC is then obtained using the Burg method (with an eighth-order autoregressive model) and a bandpower quantity computed by integration over one of four frequency bands. In our implementation, we estimated bandpower for the FMC, VC, lSMC, and rSMC ICs using the respective bands 4-8 Hz (theta), 8-12.5 Hz (alpha), 10-12.5 Hz (mu), 12.5-30 Hz (beta). This

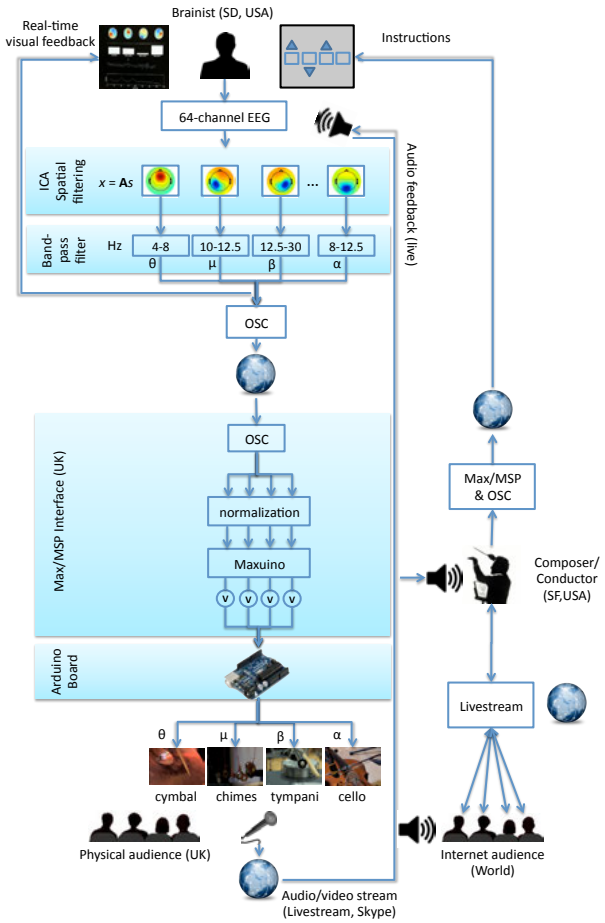


Figure 1. Installation flowchart for *Music for Online Performer*. Globes represent Internet transmission.

choice was informed by a large quantity of published literature relating power modulation in these bands, near the four selected brain areas, to several mental tasks such as motor imagery, mental calculation, and relaxation. Specifically, it is known that motor imagery (imagination of body part movement) leads to a decrease in mu and beta power, termed event-related desynchronization (ERD), in the region of sensorimotor cortex corresponding to the body part being imaged with a concomitant increase in power (event-related synchronization (ERS)) in distal regions of sensorimotor cortex. Relaxation is known to result in alpha ERS in visual cortex, while visual imagery or task engagement/focus leads to alpha ERD. Engagement in tasks with high working memory demands, such as mental calculation, is associated with increases in frontal midline theta power [12,13].

The four bandpower estimates are then fed back to the user via a bar graph display. Such real-time neurofeedback is known to be a powerful tool in improving the ability of an individual to modulate his/her neuronal rhythms and is considered an integral component of a closed-loop BCI [14]. The same bandpower estimates are also simultaneously packaged and transmitted to a computer at the performance site (Phipps Hall, University of Huddersfield, United Kingdom) using Open Sound Control (OSC).

2.3 Acoustic Instrument Control

At the performance site, OSC packets are unpacked and imported into Max/MSP, where the bandpower values are rescaled, converted into servomotor angular rotation values,

Figure 2. Instruments used in Music for Online Performer

and transmitted to an Arduino board over an RS232 (serial) interface using the Maxuino patch developed by Chris Coleman[1]. The Arduino board (we used the Arduino Duemilanove with the ATmega168 microcontroller) uses a mixture of analog and digital pulse-width modulation (PWM) sequences to control four servomotors, each of which mechanically manipulates a separate musical instrument thereby acoustically sonifying the respective bandpower. The four instruments chosen, with respective frequency band / brain anatomy mappings were cello (alpha, VC), chimes/bells, (mu, lSMC), woodblock (beta, rSMC), and cymbal (theta, FMC). The instruments were chosen for their percussive quality (with a nod towards Lucier's own choice of percussive instruments in *Solo Performer*) as well as based on our ability to effectively manipulate the instrument using a simple rotational servomotor. The mechanical devices actuating the instruments (shown in Figure 2) were designed as follows.

The cello, using standard A3/D3/G2/C2 tuning, was played via a cello bow attached to a mechanical 'arm' which connected to a rotational servomotor the angle of which was smoothly varied between 45 and –45 degrees by a 4 Hz oscillator. This produced a "tremolo" effect. The specific note evoked by the tremolo was determined via the brainist's alpha power modulation. Alpha power was scaled to the range [0 90] degrees and added as an offset to the servomotor angle. This changed the mean angle the bow made with the cello neck producing a bowed tremolo over a different subset of strings.

The chime array was actuated by a 9V DC fan whose speed varied inversely proportionate to mu power. The chimes (an array of 20 washer discs ranging in size and weight) were distributed from heaviest to lightest (front to rear) such that increases in fan speed (due to mu ERD) would resonate the heavier chimes resulting in an overall higher pitch effect.

The woodblock was actuated by a double ball-headed drumstick attached at its midpoint to a servo with a 180 degree angular range and positioned over the woodblock. Similar to mu, beta power was inversely mapped to rotation speed such that beta ERD (as occurs in motor imagery) would lead to increased percussive tempo.

The cymbal was actuated by a standard drumstick attached to a 360 degree full-rotation servo via a piece of string and positioned over an upturned cymbal. The angular velocity of the servo was varied proportionately to theta power. This produces a continuous "sweeping" or oscillating timbre whose volume can be varied by modulating the rotational velocity of the servo; increasing the rotational velocity causes the drumstick to brush the cymbal at a higher rate, increasing the resonance of the cymbal and thus the perceived volume.

The frequency-instrument mappings were selected so as to map the more controllable frequencies (respectively, alpha, mu, beta) to the more acoustically salient instruments in the ensemble. Additionally, the mappings were intended to loosely reflect the acoustic qualities of the individual neural frequencies. For instance, the rhythmic sweeping sound of the cymbal was evocative of low-frequency "droning" of a 3-7 Hz theta rhythm while the rapid beating and sharp attack of the woodblock was evocative of the high-frequency beta rhythm.

2.4 Audience Participation and Feedback

In our installation, a live audience in Phipps Hall observed the performance first-hand. Simultaneously, live audio and video (from all three geographic locations) was recorded and streamed over the Internet using freely available software (here, Skype and Livestream) to a virtual global audience. Here we had a public Livestream channel/chatroom setup, which audience members could log in to and communicate with each other and the conductor while watching the live performance.

A branch of the audio stream was transmitted to a composer/conductor in another location (here, San Francisco, USA). The conductor had a Max/MSP control interface, which was linked via OSC to the brainist's compositional instructions/notes display, implemented in Matlab. Based on a predetermined, loosely structured, compositional score and the influences of the audience, the conductor could direct the brainist to individually modulate different instruments (e.g., increase the cello pitch by increasing alpha bandpower through relaxation).

A third branch of the audio stream was fed back to the brainist who could use this, along with visual neurofeedback, to help control his neuronal rhythms. This also allowed the brainist to experience the full musical ensemble, making the BCI-instrument interaction less abstract and affording an element of direct improvisational control in the ongoing evolution of the composition.

3. DISCUSSION

Music for Online Performer was a novel venture in several regards. Perhaps the most important novel element was our use of acoustic media, with instruments actuated by low-cost Arduino robotics hardware. This stands in contrast to the majority of BCMIs that have used digital/synthesized audio as their primary media. The use of acoustic instruments introduces an additional element of uncertainty in performances, which we believe is important for compositional expressiveness. Nuances of the performer's modulation of his or her neural state may result in unpredictable behavior of the instruments, due to the nature of their physical construction. How far one attempts to mentally compensate for this unpredictability is a measure of one's willingness to "let go" of a perfect rendition and leave elements to chance.

Secondly, performances and installations combining BCMI technology and synthesized music can be somewhat abstract and acousmatic in nature. Even when the performer is visible, he or she is often immobile and the mechanism of sound production is unclear. This form of musical expression may alienate some audiences, as there is no immediate physicality to the sounds they are hearing. Using acoustic instruments allows the audience to engage with a method of sound production familiar to them and then move on to trying to understand how these instruments are being controlled.

[1] http://www.maxuino.org/

Aside from the novelty of controlling musical instruments 4000 miles away using one's thoughts, *Online Performer* was also in many ways a social experiment. By allowing audience members from around the globe to be brought together in a virtual space where they could communicate with each other throughout the performance, and influence the ongoing composition through their live interactions with the composer, we sought to highlight new kinds of social environments for musical performance. By encouraging audience participation in the physical musical production we effectively extended the virtual space back into the real and tangible, which, as Marshall McLuhan discusses in his 1994 book *Understanding Media: The Extensions of Man,* is the opposite of what usually happens with technology.

4. CONCLUSIONS AND THE FUTURE

In this paper we reported the live demonstration of a novel Internet-enabled acoustic brain-computer music interface system. To our knowledge, this is the first BCMI that has attempted to mechanically control acoustic instruments over the Internet using non-invasive EEG and low-cost, off-the-shelf Arduino robotics hardware, accessible to most artists and do-it-yourself hobbyists. Although we used medical-grade EEG equipment, affordable, high-quality EEG hardware is now becoming ubiquitous with a number of companies offering dry (gel-free) electrode systems (BrainProducts, Emotiv, Quasar, g.Tec, Nouzz, Neurosky, etc)

Although EEG is not a novel element in the experimental arts, it is only recently, with the advent of low-cost wearable EEG hardware, exponentially increasing computing capabilities, and powerful new signal-processing algorithms from the expanding neuroscience and BCI fields, that we are seeing a renewed interest in and expansion of the applications of EEG technology in the arts. As our knowledge of human cognitive neuroscience increases and low-cost EEG technology advances and becomes ubiquitous, we will see a new generation of artists, technologists, and musicians with a passion for artistically representing and expressing the subtle nuances and inner workings of the human mind via the use of brain-machine interfaces. At the same time, there will be a rise in the number of for-profit companies aimed at this generation of DIY bio-artists. Currently, one such company – InteraXon – has gained worldwide recognition for its development of BCI-enabled artistic performance pieces, including lighting up the CN Tower, Ottawa Parliament Buildings, and Niagara Falls at the 2009 Winter Olympics using wearable EEG (Neurosky's MindSet™) with brainwaves streamed from Vancouver.

As BCI technology develops, we may one day be able to remove the boundary of sensorimotor input/output and directly communicate our intentions, emotions, and desires to machines and human beings in our surrounding environment as well as across the globe. The effect will be extension of the neurobiological networks underlying thought and body schema representation and expression into much larger, externalized networks encompassing multiple other conscious and nonconscious agents.

In producing *Music for Online Performer* we found a beautiful poetry in the ubiquity and interplay of multi-scaled internalized and externalized networks and loops. On some levels of description, micro- and macroscopic neurobiological networks in the brain of the performer were rapidly transmitting information, translating the conductor's instructions into cognitive thought processes which manifested as detected modulations in neural activity influencing his local feedback display and thereby again his neural processes. Simultaneously, on other levels of description, this same neural information was being routed through megascopic globe-spanning networks, creating live acoustic music halfway around the world, influencing the neurobiological networks – and thereby the perceptions, emotions, and intentions – of others worldwide, and ultimately returning, via the directives of the audience and the human conductor, to again influence the source: a solo performer sitting in a room; alone, yet intimately connected to the world at large.

5. ACKNOWLEDGMENTS

R.W. is supported in part through Subito, the quick advancement grant program of the SF Bay Area Chapter of the American Composers Forum. Many thanks to Chris Coleman who consulted with R.W. on the Maxuino interface. T.M. thanks Yijun Wang and Nima Bigdely-Shamlo for their help with configuring the EEG system. Finally, a big thanks to all those who participated as audience members and helped create the interactive experience.

6. REFERENCES

[1] Adrian, E.D. and Matthews, B.H.C. The Berger Rhythm: potential changes from the occipital lobes in man. *Brain 57*, 4 (1934), 355.

[2] Bell, A.J. and Sejnowski, T.J. An information-maximization approach to blind separation and blind deconvolution. *Neural comp. 7*, 6 (1995), 1129–1159.

[3] Brouse, A. The Interharmonium: an investigation into networked musical applications and brainwaves. M.A Thesis. (2001).

[4] Delorme, A., Mullen, T., Kothe, C., et al. EEGLAB, SIFT, NFT, BCILAB, and ERICA: New tools for advanced EEG/MEG processing. *Computational Intelligence and Neuroscience,* In Press, (2011).

[5] Grierson, M. Composing With Brainwaves: Minimal Trial P300 Recognition As An Indicator of Subjective Preference for the Control of a Musical Instrument. *ICMC*, (2008).

[6] Hinterberger, T. and Baier, G. Parametric orchestral sonification of EEG in real time. *Multimedia, IEEE 12*, 2 (2005), 70–79.

[7] Lucier, A. Interview: Everything Is Real. *TAG Publishing*, 2010.

[8] Mann, S., Fung, J., Garten, A. DECONcert: Bathing in the light, sound, and waters of the musical brainbaths. *ICMC*, (2007) Copenhagen.

[9] Miranda, E.R. and Brouse, A. Interfacing the Brain Directly with Musical Systems: On Developing Systems for Making Music with Brain Signals. *Leonardo 38*, 4 (2005), 331-336.

[10] Miranda, E.R. and Wanderley, M. *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. A-R Editions, 2006.

[11] Makeig, S., Bell, T., Jung, T., Sejnowski, T. Independent component analysis of electroencephalographic data. *NIPS 8*, (1996), 145–151.

[12] Pfurtscheller, G. and Lopes da Silva, F.H. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clin. Neurophysiology 110*,11(1999), 1842-57.

[13] Pfurtscheller, G. Functional topography during sensorimotor activation studied with event-related desynchronization mapping. *Clinical Neurophysiology 6*, 1 (1989), 75-84.

[14] Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., and Vaughan, T.M. Brain-computer interfaces for communication and control. *Clinical Neurophysiology 113*, 6 (2002), 767–791.

[15] Wu, D., Li, C.-Y., and Yao, D.-Z. Scale-free music of the brain. *PloS one 4*, 6 (2009), e5915.

Rhythm'n'Shoes: a wearable foot tapping interface with audio-tactile feedback

Stefano Papetti
Dept. of Computer Science,
University of Verona
Strada Le Grazie, 15
37134 Verona, Italy
stefano.papetti@univr.it

Marco Civolani and Federico Fontana
Dept. of Mathematics and Computer Science,
University of Udine
Via delle Scienze, 206
33100 Udine, Italy
name.surname@uniud.it

ABSTRACT

A shoe-based interface is presented, which enables users to play percussive virtual instruments by tapping their feet. The wearable interface consists of a pair of sandals equipped with four force sensors and four actuators affording audio-tactile feedback. The sensors provide data via wireless transmission to a host computer, where they are processed and mapped to a physics-based sound synthesis engine. Since the system provides OSC and MIDI compatibility, alternative electronic instruments can be used as well. The audio signals are then sent back wirelessly to audio-tactile exciters embedded in the sandals' sole, and optionally to headphones and external loudspeakers. The round-trip wireless communication only introduces very small latency, thus guaranteeing coherence and unity in the multimodal percept and allowing tight timing while playing.

Keywords

interface, audio, tactile, foot tapping, embodiment, footwear, wireless, wearable, mobile

1. INTRODUCTION

In many cultures, music and dance performers make use of foot tapping, from folk fiddlers and street buskers to flamenco and tap dancers. For instance, a fiddler stomping on a pub's wooden floor can cheer on the audience meanwhile supporting his or her own playing by adding a simple percussion part; buskers often include foot drums in their setup to add even complex percussion parts to their guitar playing. Moreover, traditional musical genres exist where players make extensive use of foot percussions (*podo-rhythm*) as main accompaniment. As for dance, foot tapping can have both an expressive and rhythmic function, to the extent that some dance genres are centered on the musical and gestural performance produced by the dancer's feet.

On the other hand, in everyday life many musicians and music enthusiasts alike find themselves "tapping songs" with their fingers, hands and feet. Such tapping may represent the song's main melody, rhythm, or even accurately simulate its percussions part.

The gesture of playing rhythms with the feet offers spontaneity and expressivity, at the same time enabling an embodied experience.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

Taking inspiration from these observations, and starting from a prototype shoe-based interface we had previously realized for interactive walking purposes [18], we implemented a wearable controller for foot tapping that we have named "Rhythm'n'Shoes". A peculiarity worth noticing is that the interface provides the user with foot-level audio-tactile feedback through exciters embedded in the shoes' sole.

A similarly immediate approach to playing rhythms that avoid the use of virtual drum interfaces, but instead takes inspiration from the common experience of hitting the chest or thighs with the hands, is depicted in [2]: the interface consists of a pair of gloves embedding piezo microphones that are used as sensing devices. Several recent studies take into account novel percussion instruments [1, 6] and interfaces for percussion tasks [9, 25].

As for foot-based interfaces, various works exist that describe instrumented shoes and floors for interactive dance [21, 19] or other musical purposes [16, 11]. Such examples present higher latencies and lower sampling rates compared to our prototype (see Section 2.1). Moreover, those interfaces only act as controllers tracking the user's gestures, while they do not directly provide any feedback. A few notable exceptions out of a musical context are [23, 24], where foot-level haptic feedback is provided.

Various researches consider the use of haptic feedback in digital musical interfaces and instruments [5, 17, 15]. With regard to interfaces for percussion tasks, haptic feedback is exploited in e.g. [12] and [4].

2. INTERFACE DESIGN

This section describes the design of the interface from the hardware implementation to the software level.

2.1 Hardware

Starting from the top left of Fig. 1, a pair of sandals are equipped with four force sensing resistors (Interlink 402 FSR) fixed under the insole, one at the toe and one at the heel. The FSR sensors are connected to the analog inputs of an Arduino Duemilanove board (*force data transmitter*). Here the force signals are sampled and encapsulated using a custom protocol [7] and sent to a 2.4 GHz wireless transceiver module based on the nRF2401A chip by Nordic Semiconductor. A one-directional wireless line is realized by connecting a specular system: another nRF2401A module receives the data stream and routes it to an Arduino board (*force data receiver*). The latter is interfaced via a USB connection with a personal computer running Pure Data (Pd). Here the received data are processed to generate audio-tactile signals[1] to be sent to the sandals (see Section 2.2).

[1]As described in Section 2.2, while the system can be directly interfaced with MIDI and OSC compatible instru-

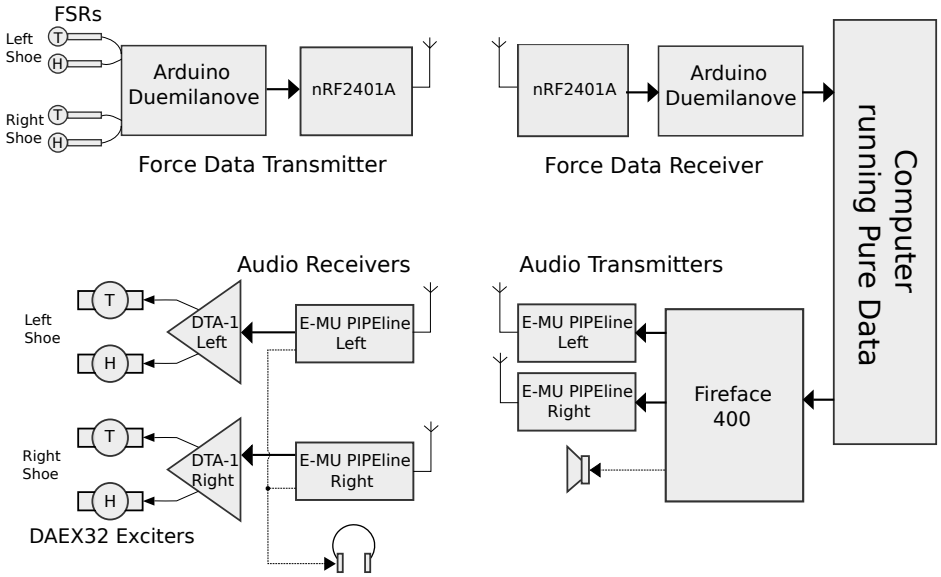


Figure 1: Block diagram representing the low-level hardware setup. The upper and lower flows illustrate respectively the sub-systems implementing force data acquisition and audio-tactile feedback.

Each sandal embeds two exciters that are driven by the outputs of a RME Fireface 400 multichannel audio interface. In detail, starting from the bottom right of Fig. 1, four output channels of the RME are grouped into two stereo pairs and each pair is routed to an E-MU PIPEline wireless audio transceiver. Each E-MU on the computer side (*audio transmitters*) is paired with another one on the user side (*audio receivers*), thus obtaining a four-audio-channel simplex wireless connection. Finally, the outputs of each E-MU receiver are injected into a Dayton Audio DTA-1 stereo amplifier which drives two Dayton Audio DAEX32 exciters fixed under each sandal – one at the toe and one at the heel – thus closing the interaction loop.

Since the setup is conceived for live performance, freedom of movement must be ensured, and therefore wireless communication represents an ideal choice. On the other hand the overall latency must be kept as low as possible, and the interface needs to be wearable (i.e. lightweight and small). The whole hardware system has been designed to satisfy such requirements, moreover using readily available components. At the user side, one Arduino with the attached nRF2401A module (*force data transmitter*), two E-MU units (*audio receivers*) and the two DTA-1 amplifiers are carried into a small backpack worn by the performer, together with standard batteries. Each sandal is then connected to the backpack via a single multi-conductor cable, this way minimizing encumbrance. The round-trip latency exhibited by the system – measured as the delay between the onset of an impulse at the FSR sensors and the arrival of the corresponding feedback signals to the exciters [7] – amounts to about 20 ms.

2.1.1 Details on data acquisition

Several solutions for musicians and performers have already been proposed which offer wireless acquisition of control signals[10, 22, 8], however most of them are based on custom hardware and/or are quite expensive. In our prototype, on the contrary, the wireless transmission of force data is managed by two readily available and low-cost transceivers based on the nRF2401A chip. Moreover our choice of developing a custom data protocol and send it over a dedicated wireless connection was necessary to avoid the latency and sampling rate drawbacks that other standard solutions (e.g. WiFi, Bluetooth or ZigBee) would have imposed.

Despite its low cost, the Arduino Duemilanove offers high-performance signal acquisition functionalities [7]. We have configured its microcontroller's ADC to uniformly sample up to six analog channels with a fairly high-rate and 10 bit resolution: the sampling frequency per single analog channel depends both on the serial data rate and number of channels [18]. With four channels, as in our case, the resulting frequency is 1050 Hz per single channel.

The latency introduced by the data acquisition system amounts to about 1.2 ms.

2.1.2 Details on feedback

The interface provides the performer with four-channel audio-tactile feedback:two E-MU PIPEline are used to send four audio signals introducing a delay of 5.5 ms (from official specifications).

The used exciters are meant to generate audio-rate vibrations, therefore abundantly covering the bandwidth required for haptic display [15].

2.2 Software

At the software level, three modules realized in Pd are organized in a bottom-up hierarchy: 1) at the first layer, the data stream generated by the FSR sensors is conditioned and analyzed in order to detect tapping events. As soon as one of such events is detected, this module outputs a measure of its energy; 2) the second layer maps the detected events alternatively to MIDI or OSC messages, or directly to the parameters of a sound synthesis engine running in Pd; 3) the third and last layer implements a physics-based impact sound model, which is driven by the detected tapping events. These three layers are described in detail below.

2.2.1 Data conditioning and analysis

The force data are received and unpacked, this way obtaining four separate streams respectively corresponding to the four FSR sensors (left/right heel and toe).

These streams are then conditioned and optimized in view

ments, the prototype already provides a synthesis engine implemented in Pd, this way offering a self-contained setup. For the sake of simplicity, in what follows we refer to the included synthesis engine.

of the following processing stage: the data is passed through threshold gates to filter out signal noise and avoid unwanted rebounds in the impact detection process.

The pre-conditioned force data are then analyzed in order to detect the onset of tapping events and measure their energy. To this end we made use of a Pd object called `bonk~` [20], which decomposes the incoming signal into frequency bands and computes the power in each of them, then it looks for sharp edges in the spectral envelope of the signal, enabling a very accurate detection of percussive events. As `bonk~` is meant to analyze audio signals, before being sent to it the pre-conditioned data streams are converted accordingly: they are oversampled from the original sample rate of 1470.5 Hz to Pd's internal audio rate by using the Pd object `sig~`. The resulting signals are then processed by a simple anti-aliasing filter. The output provided by `bonk~` consists in a measure of the energy of the detected event, calculated as the sum of the square roots of the amplitudes in each frequency band.

2.2.2 Mapping

The energy values of the detected tapping events are used to drive the control parameters of different instruments. In particular, a threefold path has been implemented, complying with three distinct protocols:

MIDI: the energy values are converted into integer values in the range 0-127 to comply with MIDI velocity values. As soon as a tapping event is detected, a "note on" message is generated and associated with such velocity value. A "note off" message is produced following each "note on" message, after a settable delay time. The "channel" and "note number" for each of the four data streams can be assigned to interact with any MIDI instrument, however the default configuration already offers a common drum setup according to the General MIDI standard.

OSC: since OSC-compatible instruments require custom messages, the user can modify the generated OSC messages to taste. For example using Pd's object `maxlib/scale` the original range of energy values (0-100) can be converted to any range of choice. The default configuration already offers predefined messages for communicating the onset of tapping events and their energy, while energy values are expressed as floating-point numbers from 0 to 100.

SDT: energy values are converted into physically-consistent velocity values expressed in m/s, that are sent to a physics-based impact sound model.

The three mappings described above can be selected alternatively via a switch implemented in Pd.

2.2.3 Sound synthesis

In order to provide a self-sufficient system, a sound synthesis engine was included in our prototype, this way transforming the interface into a complete instrument.

The sound synthesis engine makes use of a physics-based impact model [3] which is part of a library for Max and Pd called Sound Design Toolkit (SDT).[2] The model simulates a mass (object 1) colliding with a resonator (object 2), and the model's output represents the vibrations of the latter. Therefore the synthesized signals are particularly suitable to drive both audio and vibrotactile feedback.

In more detail, the contact between the two objects is accounted for by a nonlinear spring with dissipation, while the resonator is modeled according to the modal synthesis paradigm. The available control parameters give access to: the mass m (in Kg) of object 1; the resonating modes of object 2, namely their frequencies $f_{0..n}$ in Hz (where n is the number of modes), their decay times $t_{0..n}$ in s, and their gains $g_{0..n}$; the nonlinear spring, namely its nonlinearity exponent α and its stiffness k in Kg/N$^{\alpha}$. Such parameters enable the user to design sounds that simulate a wide variety of object's sizes and materials, like wood, plastic, metal and glass.

Each force data stream is mapped to a separate instance of the impact model, resulting in a different sound for each tapping position.

[2]Freely available at `http://www.soundobject.org/SDT`.

Figure 2: A performer wearing the interface, tapping the feet while sitting.

3. THE INTERFACE IN USE

The system has currently been calibrated for playing in a sitting position (see Fig. 2), which minimizes the detection of spurious tapping events. On the contrary, the calibration required for playing while standing up is obviously trickier, as the performer inevitably has to adjust his/her posture, e.g. to balance.

Thanks to Velcro straps, the sandals can easily fit a wide range of foot sizes, both bigger and smaller than their native European size 44 (corresponding to U.S. male size 10 1/2).

As shown in Fig. 1 the user can connect headphones and/or external loudspeakers to the interface, e.g for rehearsing purposes or for performing on stage.

Although the system is especially suited to play percussion instruments, it is not just limited to them. Indeed the availability of MIDI and OSC controls on the one hand allows to connect the interface to potentially any electronic or computer-based instrument, on the other hand it enables the implementation of complex mappings for supporting the experimentation of further musical styles and aesthetics.

Digital musical instruments usually lack the tactile feedback that is inherently conveyed by most traditional instruments. Such vibrations stimulate the mechanoreceptors in the skin [15]: in particular, the fingers are sensitive to vibrations up to 1000 Hz with a peak at about 250 Hz, and while it is generally acknowledged that the foot is less responsive than the hand, similar sensitivity figures are found for the foot sole [13]. Sensitivity thresholds also depend on the area of contact and the nature of the stimuli.

As explained in Section 2.1, the exciters embedded in the sole are driven by audio signals, therefore the resulting vibrotactile feedback ensures a tight coupling with the action

of tapping. Informal evaluation done while testing the interface showed that such energetic consistency gives rise to a fairly convincing experience: in particular, by using the included physically-consistent impact model both the audio and tactile feedback improve on dynamics and realism.

Despite the fact that a maximum of 10 ms latency is generally suggested for music controllers [14], from informal evaluations we have found that our system is very responsive, and guarantees coherence and unity in the multimodal percept (see Section 2.1 for the measured latency figure). This is possibly partly due to the fact that the feet are not as sensitive as the hands, thus resulting in a higher tolerance to foot-level delays. As a result, the user is able to play with remarkable accuracy even fast paced and complex rhythms.

Also, tests showed that the implemented wireless communication is solid and reliable, independently of the performers' movements and within a range of about 15 meters.

Since the interface does not require any visual skill and provides vibrotactile feedback, it is perfectly suitable for both blind and hearing impaired persons. For example, as an alternative usage the hearing impaired could exploit the interface's feedback as a personal monitoring system, especially effective for feeling rhythmic parts or just the tempo.

4. CONCLUSIONS

The act of tapping the feet to play rhythms guarantees spontaneity and expressivity, while allowing the skilled performer to use more than one instrument (or other devices) at a time.

The "Rhythm'n'Shoes" interface is suitable for traditional musical genres where players make use of foot drums, as well as other types of performance where enhanced control over electronic (e.g. MIDI or OSC compatible) devices is required.

Even if proper testing is still needed, preliminary informal evaluation shows that the system exceeds by far the expressivity offered by simple trigger-based interfaces. Furthermore, thanks to the provided audio-tactile feedback, the interface offers a truly embodied interaction, even while playing an electronic instrument. Additionally, the use of physics-based sound models for generating both the audio and tactile feedback provides a consistent and realistic experience.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme under FET-Open grant agreement 222107 NIW - Natural Interactive Walking.

6. REFERENCES

[1] R. M. Aimi. New expressive percussion instruments. *M.S. Thesis. MIT*, 2002.

[2] M. S. Andresen, M. Bach, and K. R. Kristensen. The lapslapper: feel the beat. In *Proc. Int. Conf. on Haptic and Audio Interaction Design (HAID)*, pages 160–168, Berlin, Heidelberg, 2010. Springer-Verlag.

[3] F. Avanzini, M. Rath, and D. Rocchesso. Physically-based audio rendering of contact. In *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, volume 2, pages 445–448, 2002.

[4] E. Berdahl, W. Verplank, J. O. Smith III, and G. Niemeyer. A physically intuitive haptic drumstick. In *Int. Computer Music Conf. (ICMC)*, page 363–366, Copenhagen, Denmark, 2007.

[5] C. Chafe. Tactile Audio Feedback. In *Proc. Int. Computer Music Conf. (ICMC)*, Tokyo, Japan, 1993.

[6] K. Chuchacz, S. O'Modhrain, and R. Woods. Physical models and musical controllers: designing a novel electronic percussion instrument. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, pages 37–40. ACM, 2007.

[7] M. Civolani, F. Fontana, and S. Papetti. Efficient acquisition of force data in interactive shoe designs. In *Proc. Int. Workshop on Haptic and Audio Interaction Design (HAID)*, Copenhagen, Denmark, 2010. Springer.

[8] T. Coduys, C. Henry, and A. Cont. TOASTER and KROONDE: high-resolution and high-speed real-time sensor interfaces. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2004.

[9] M. Collicutt, C. Casciato, and M. M. Wanderley. From real to virtual: A comparison of input devices for percussion tasks. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2009.

[10] E. Fléty. The WiSe Box: a multi-performer wireless sensor interface using WiFi and OSC. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2005.

[11] J. A. Hockman, M. M. Wanderley, and I. Fujinaga. Real-time phase vocoder manipulation by runner's pace. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2009.

[12] S. Holland, A. J. Bouwer, M. Dalgelish, and T. M. Hurtig. Feeling the beat where it counts: fostering multi-limb rhythm skills with the haptic drum kit. In *Proc. Int. Conf. on Tangible, Embedded, and Embodied Interaction (TEI)*, pages 21–28, New York, NY, USA, 2010. ACM.

[13] J. Kekoni, H. Hämäläinen, J. Rautio, and T. Tukeva. Mechanical sensibility of the sole of the foot determined with vibratory stimuli of varying frequency. *Experimental Brain Research*, 78:419–424, 1989.

[14] T. Mäki-Patola and P. Hämäläinen. Effect of latency on playing accuracy of two continuous sound instruments without tactile feedback. In *Proc. Int. Conf. on Digital Audio Effects (DAFx)*, pages 11–16, Naples, Italy, October 2004.

[15] M. T. Marshall and M. M. Wanderley. Vibrotactile feedback in digital musical instruments. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, pages 226–229, Paris, France, 2006. IRCAM.

[16] B. Moens, L. van Noorden, and M. Leman. D-Jogger: Syncing Music with Walking. In *Proc. Sound and Music Computing conf. (SMC)*, volume online, pages 451–456, Barcelona, 2010. Universidad Pompeu Fabra.

[17] M. S. O'Modhrain. *Playing by feel: incorporating haptic feedback into computer-based musical instruments.* PhD thesis, Stanford, CA, USA, 2001.

[18] S. Papetti, F. Fontana, M. Civolani, A. Berrezag, and V. Hayward. Audio-tactile display of ground properties using interactive shoes. In *Proc. Int. Workshop on Haptic and Audio Interaction Design (HAID)*, Copenhagen, Denmark, 2010. Springer.

[19] J. Paradiso, K. yuh Hsiao, and E. Hu. Interactive music for instrumented dancing shoes. In *Proc. Int. Computer Music Conf. (ICMC)*, pages 453–456, 1999.

[20] M. Puckette, T. Apel, and D. Zicarelli. Real-time audio analysis tools for Pd and MSP. In *Proc. Int. Computer Music Conf. (ICMC)*, pages 109–112, Ann Arbor, Michigan, USA, 1998.

[21] P. Srinivasan, D. Birchfield, G. Qian, and A. Kidané. A pressure sensing floor for interactive media applications. In *Proc. 2005 ACM SIGCHI Int. Conf. on Advances in Computer Entertainment Technology*, ACE, pages 278–281, New York, NY, USA, 2005. ACM.

[22] D. Topper and P. V. Swendsen. Wireless dance control: PAIR and WISEAR. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2005.

[23] Y. Visell, A. Law, and J. R. Cooperstock. Touch is everywhere: floor surfaces as ambient haptic displays. *IEEE Transactions on Haptics*, 2009.

[24] J. Watanabe, H. Ando, and T. Maeda. Shoe-shaped interface for inducing a walking cycle. In *Proc. Int. Conf. on Augmented Tele-existence*, ICAT, pages 30–34, New York, NY, USA, 2005. ACM.

[25] D. Young and I. Fujinaga. AoBachi: A new interface for Japanese drumming. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2004.

A structured design and evaluation model with application to rhythmic interaction displays

Cumhur Erkut, Antti Jylhä
Aalto University
School of Electrical Engineering
Dept. Signal Processing and Acoustics
PO Box 13000 Aalto FI-00076, Finland
cumhur.erkut@aalto.fi, antti.jylha@aalto.fi

Reha Dişçioğlu
Aalto University
School of Art and Design
Media Lab
PO Box 31000 Aalto FI-00076, Finland
reha.discioglu@aalto.fi

ABSTRACT

We present a generic, structured model for design and evaluation of musical interfaces. This model is development oriented, and it is based on the fundamental function of the musical interfaces, i.e., to coordinate the human action and perception for musical expression, subject to human capabilities and skills. To illustrate the particulars of this model and present it in operation, we consider the previous design and evaluation phase of iPalmas, our testbed for exploring rhythmic interaction. Our findings inform the current design phase of iPalmas visual and auditory displays, where we build on what has resonated with the test users, and explore further possibilities based on the evaluation results.

Keywords

Rhythmic interaction, multimodal displays, sonification, UML

1. INTRODUCTION

Structured approaches in design and evaluation of novel musical interfaces are rare. Even rarer are the cases that build on the evaluation of the previous design phase, and implement the insights gained from user observations in the next phase. There is a clear need for such cases if deployment is desired, to understand how the intentions of designers are perceived and utilized by the users.

Currently, the purpose and function of evaluation of musical interfaces are in focus within the NIME community [6]. While our knowledge on musical perception, cognition, and interaction is rapidly advancing, there is a lack of practice of describing which capabilities are addressed in design, how various aspects are constraining the utilization of these capabilities, and how the mappings between the human capabilities and computational modalities are aligned. Similar observations were reported in [5, 4] regarding multimodal interfaces, and a structured approach has been proposed.

In this paper, we are primarily interested in repurposing this model for NIME. We first explain this structured approach and the corresponding design and evaluation models in Sec. 2. We then frame the previous design and evaluation phase of *iPalmas*, our testbed for designing and evaluating rhythmic interaction [2], within this model in Sec. 3. We build on all of these to present our ideas for the next design phase of iPalmas in Sec. 4. We finally derive our conclusions and indicate our future work in Sec. 5.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. DESIGN AND EVALUATION MODEL

The basic idea of our approach, illustrated in Fig. 1, is that multimodal interactive systems are designed to coordinate the human action and perception for a particular effect, subject to human capabilities and skills. Several constraints may break the design intentions in deployment. The model structurally decomposes the computer modalities, human capabilities, and evaluation issues in a way similar to how *Unified Modeling Language* (UML) structures a modeling domain. UML is a generic computational modeling approach in software development, in which the focus and primary artifacts of development are the *models* instead of *programs* [3]. The main goals are to understand the domain, express the solution in various abstraction levels in the form of *structural* and *behavioral* diagrams, and evaluate in the realm of models and prototypes.

2.1 Design model

Fig. 1 illustrates the multimodal interaction model based on UML profiling and extensions. The model is a synthesis of input and output modalities, and their integration, expressed however in the UML framework. Profiling means that the model is specialized for a particular domain, and extension means that it includes special modeling elements.

The model is based on the practical definition of multimodal systems for musical interaction, consisting of an interface and supporting application that aim to *produce* a particular *effect* on a user, with parameters shared by this effect and a (computational) modality. The effect can be sensory, perceptual, motor, or cognitive, often forming a hierarchy by causality: the perceptual effects are usually based on sensory effects, etc, all the way up to human musical capabilities.

The musical interface can employ a *simple modality*, for instance visual or auditory, or multiple modalities by integrating simple modalities, such as audio-visual, or audio-tactile. In this case, we talk about *complex modalities*. The *multimodal integration* can be done sequentially or concurrently, always within a *time-frame*. From the computer point of view, we acquire *input modalities* with sensors, e.g., microphones, or accelerometers, or input devices. Some input devices are *event-based* (a key-press or a mouse-click), while most sensors provide continuous data streams by sampling. *Streaming-based modalities* are always indicated by their sampling frequency attribute. These are specialized as *recognition-based modalities*, which are specified by a recognition *error-rate* attribute.

In some cases, a recognition-based modality can convert a streaming-based modality into an event-based modality.

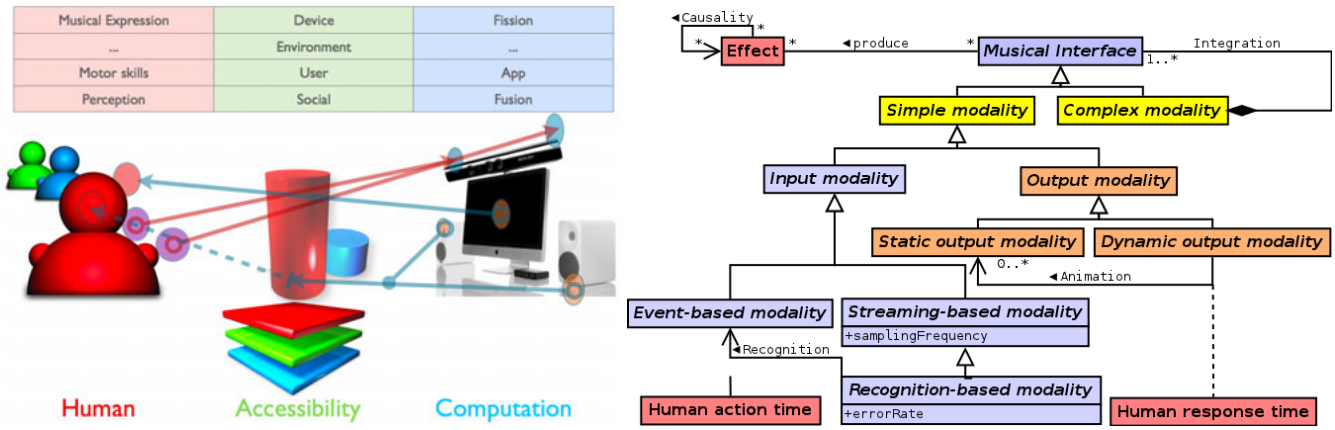


Figure 1: The structured design and evaluation model we are following, after [4] (left), and the corresponding multimodal interaction model (right). The human capabilities, computational modalities and accessibility issues are expressed as structured layers in this framework, where the latter is utilized to check how well the intentions of design are perceived by the user.

For instance, we can perform a percussive event recognition and classification on an audio stream, where the mainstream algorithms yield an error rate of 20 %. In computing and rendering the *output modalities*, we always consider the human response and its time-scale.

The model specializes the output modality as static or dynamic, with an *animation* association between the modalities, which is constrained by the *human response time-scale*. The human interactive response is considered at three levels: perceptual processing (about 0.1 second), immediate response (about 1 second), and unit task (about 10 seconds). For instance, the animation of static images is considered to produce a movie with smooth motion, if the duration of each image is less then the perceptual processing response time. For rhythmic interaction, the perceptual processing time of a *smear window* is important to perceive the event order, as a necessity of cognitive function [1].

2.2 Evaluation model

Similarly, the evaluation constraints can also be structurally decomposed in basic and complex constraints, and two main types of basic constraints can be identified: user and external constraints. The user constraints are user feature, user state (emotional and cognitive contexts), and user preference, whereas the external constraints are structured as device constraint, environmental constraint, and social context. The observations, remarks, and the evaluation outcomes then can be tabulated, similar to Fig. 1, left. Not all aspects may be evaluated in a single session, but they should still be kept in mind when designing the tests, and inference should be sought from the test results.

3. RHYTHMIC INTERACTION IN IPALMAS

iPalmas was developed for observing the rhythmic interaction of people with a maximally simple interactive system, to teach a novice user Flamenco hand clapping patterns [2]. We expect this type of interaction to engage people, without requiring any special skills. In the following, we elaborate the relation between the modeling framework presented in Sec. 2 and the design and evaluation of iPalmas.

3.1 iPalmas design model

iPalmas is designed for interaction between the user and a virtual tutor. The primary *input modality* is an audio stream of the user's performance. In producing this stream,

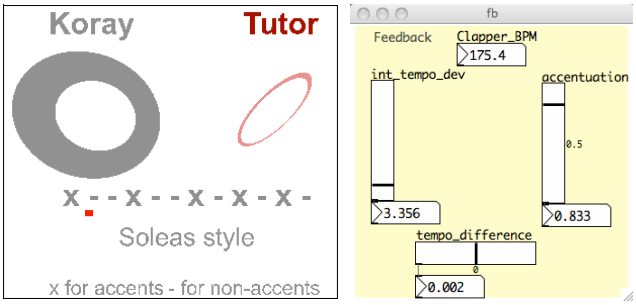


Figure 2: Three different visual displays of iPalmas. (Left) The circles to indicate the match between the tutor and the performer and the compas presentation. (Right) Various metrics presented by sliders and number boxes.

the user claps her hands in alignment with the virtual tutor, coordinates her motor action, and experiences an audio-tactile feedback naturally occuring during the clapping. As we know from sensorimotor synchronization studies, this multimodal feedback has implications on the timing of motor action [1].

The *streaming-based modality* of the user's clapping is converted to an *event-based modality* by real-time hand clap sound *recognition*. Each event is characterized by the event time, the detected hand configuration (cupped vs. straight hands), the detected accentuation (loud vs. soft claps), an update to the clapping tempo estimate, and an update to the estimate of temporal deviation in the clapping. While tempo and temporal deviation are continuous measures, they are updated only for each detected event. The system, as a *musical interface*, aims at coordinating this input modality with a *complex output modality* to present the user a Flamenco pattern to practice and an online evaluation of the user's learning and performance.

To achieve this, iPalmas utilizes both the *auditory* and *visual output modalities*. The target hand clap pattern is presented by synthetic hand clapping sounds within a reverberant environment (both *dynamic output modalities*), and a visual transcription of the accentuation (see Fig. 2, left). This transcription consists of 12 marks corresponding to the beats of a Flamenco compas (highlighted by shape, either - or x, depending on the compas), a red visual marker (highlighted by shape and color), the pattern name and the

legend of accents, all static output modalities. The visual display contains also the following four static output elements (highlighted by color): two circles and two textual elements indicating the name of the current user and the tutor. The color elements are also used for association of the circle with the text label. Here, the *perceptual effects* of *grouping by color and proximity* are in operation.

The visual *dynamic output modalities* include the animation of the visual marker and the two circles. The visual marker is animated by the tempo of the tutor, to indicate the current position within the pattern. This marker wraps to the beginning after the last beat of the compas, and a short auditory marker is played at the wrap-around. The same tempo animates the tutor circle (the right one), resetting its sway clockwise at each clap occurence. The user circle, on the left, is animated in a similar fashion, but the resets happen at each detected clap. The distance between the circles' centers gets smaller, when the user's clapping tempo gets closer to that of the tutor. At this step, we were aiming for a perceptual grouping both on *proximity* and *common fate gestalt*. The thickness of the circles indicates accentuation, with the thick circle corresponding to an accentuated clap. Finally, the circles sway clockwise and back for each (detected) clap, so when the user perfectly matches the tutor's performance, the circles move unanimously.

In addition to the abstract representation of the circles, the user is presented numeric metrics on the performance, indicating the difference between the user's and the tutor's tempo, the user's internal tempo deviation, and the incorrectness of performing the accentuation (the bottom part of Fig. 2). With perfect performance, all the metrics are zero. By using the GUI elements such as label texts, slider, and number boxes, we were aiming for *cognitive effects*.

3.2 iPalmas evaluation model

We have performed an evaluation of the iPalmas system with 16 subjects [2]. This number provided a good balance between the combinations needed by the experiment design and the discoverability of most of the usability problems with a small subject group (i.e., Nielsen's model, see [7]), in our design phase iteration.

Most of the participants had musical background, but none of them were Flamenco practitioners. They practiced four different hand clapping patterns, two with the auditory output only (hand clapping of the tutor) and two with both auditory and visual output (hand clapping, transcription, circles, and numeric metrics). In half of the cases the virtual tutor's tempo remained constant, in the rest the tempo was allowed a small drift from the original tempo, adapting to the user's clapping. In the experiment, the subjects first practiced a pattern and then performed the learned pattern for one minute without the tutor's hand clapping. The evaluation results are presented in Table 1, according to the evaluation model presented in Sec. 2.2. Since the evaluation was carried out in laboratory conditions with one subject at a time, the social and environment constraints were not tested. However, qualitative observations gathered from questionnaire and follow-up discussions provide some insights in these aspects. In the following, only the most important observations, indicated by Roman numbers in the table, will be discussed. The reader is referred to [2] for a more detailed discussion.

The auditory output was found to be the most important factor in learning the patterns (I). Out of the visual elements, the most useful one was the transcription of the pattern, with the moving marker below it (IIa,b). The rhythmic performance of the subjects varied between different pattern-tutor combinations and subjects, but in general it was found that the subjects tended to accelerate, once the auditory output faded away (III).

Some subjects showed more variation in their temporal performance than others (IV). Visual elements, namely the transcription (Va), and the allowed tempo adaptation (Vb), helped in succeeding with the accentuation. With a tempo-adaptive tutor, the time between two claps was slightly longer before an accentuated clap than before an un-accentuated clap. In general, the subjects regarded the numeric metrics (VI) and the dancing circles (VII) of limited use in the interaction and learning.

4. CURRENT DESIGN PHASE

The evaluation provided us good insights about our target group. We have considered the *user preferences* that have assessed the usefulness of auditory and visual markers, various visual elements, and especially the transcription, resulting in a new visual display, reported in the next subsection. In addition, the advanced auditory perception capabilities of some participants, who reported excessive reverberation and were disturbed by early reflections, inspired us to rely on the reverberation as an auditory display. Shortly, we are currently focusing on the audio-visual "touch-points" of iPalmas, and plan to revisit the technical aspects (device, application, system) in the next phase, finally completing the development cycle. The final design of iPalmas will be demonstrated at `http://www.acoustics.hut.fi/research/ipalmas.html`

4.1 Visual display

As observed in evaluation, having three separate graphical representations (clap pattern, metrics, and circles) did not resonate well with our subjects. A new graphical interface that unifies those three regions is under development. The concept is illustrated on Fig. 3. It is an abstraction of the traditional Flamenco compas. Note that the figure overlays several instances of visualization for brevity. The concept consists of twelve discs, arranged in a circular manner according to chosen clap pattern (Soleás in the figure). The numbers are optional, but included to stimulate the referential learning of rhythms by counting. The progress of time is represented both continuously (by a "fluid" flowing in the central, circular grey tube), and also discretely, by highlighting the position of the tutor. This highlighting can be done in several ways, either by a glow as presented in the figure on beat 3, or by simply hollowing out/refilling the particular circle, integrated with the tutor's clap.

The user activity is represented by rings, as the blue accented clap on beat 6, or the orange non-accented pattern on beat 9. When the tutor disc is highlighted simultaneously with the correct accent of the user, then a good performance is achieved. The performance indicators presented in Fig. 2 may also be used to modulate the radius of the central grey tube, in a way that the tube becomes infinitesimal when perfect performance is achieved (i.e., the performer does not need this performance measure anymore).

4.2 Auditory display and sonification

For a tight multimodal integration with the visual display, we also plan to sonify the key parameters of the user's performance. To start with, we have several parameters computed by the iPalmas system. These parameters can be divided into event-based and continuous sonification targets. The event-based targets include the correctness of each accent and the temporal offset from each of the tutor's claps. The continuous targets are tempo lead or lag (how much the user is clapping ahead/behind of the target tempo), overall

Table 1: The evaluation results of iPalmas, tabulated according to the model presented in Sec. 2.2. The social and environmental aspects, indicated by (*) are not directly observed.

	Social(*)	User	Environment(*)	Device, App
Sensory		Auditory marker distracting Visual marker distracting	Noise, masking	Cross-talk
Perceptual (Auditory)	No solo claps heard	Perceived "castanets" Reported excess reverb	Reverb	Synthetic sound Reverb algorithm
Perceptual (Visual)		Prefer static compas Audio-visual sync?		 Sync of threads
Motor	No solo claps practiced	Cannot produce accents **III. Speed up when tutor stops** Fatigue **IV. Temporal variation** **Va. Transcription helps accentuation**	Reverb	 Latency Latency
Cognitive		**I. Prefer Audio Feedback** Too many visual elements	Smear windows	Latency Threads
Memory	Comparison	2 subjects remembered all 4 patterns On average, 2 patterns remembered **IIa. Transcription helps recall**	 "Whole plus detail"	Pattern dictionary
Learning	Comparison	Prelistening (about 40 seconds) **VI. Metrics of limited use (for some)** **VII. Circles attractive, but not useful** **IIb. Transcription helps learning**	Noise, masking	Test phase design
Expression	Shared mastery	**Vb. Adaptive mode improves**	Smear windows	Critical latency

Figure 3: Concept for iPalmas visualization.

correctness in accentuation (computed with a running correctness metric), and the user's internal tempo deviation. The metrics were previously presented in numeric form, as in Fig. 2. In the sonification, we plan to concentrate on the continuous targets, as we consider them more important in continuous interaction.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a structured model, previously proposed for modeling multimodal interaction and evaluation, for the design and evaluation of musical interfaces. It can work as a tool for studying the existing design and evaluation cases, or can be used for informing the development. Here, we have utilized both on our own development, design, and evaluation of iPalmas.

While the structural decomposition of musical interfaces, interaction paradigms, and novel applications to atomic components may seem a difficult task at a first sight, we aim to build a collection of models and successful patterns [3]. Our other future task is to work out the evaluation results presented in Table 1 and complete the evaluation of social and environmental aspects. The *user features* we have observed can be summarized in a few user profiles, which may inform the next development phase. For instance, new training modes can be developed for the users who have never heard or practiced their clapping in isolation, but only in crowded concerts of similar social gatherings. On Table 1, we have correlated some user preferences with technical system components. Among them, timing, latency, and threads are crucial factors that we need to consider. Finally, we plan to evaluate the visual and auditory displays proposed in this work.

6. ACKNOWLEDGMENTS

This work is supported by the Academy of Finland (Pr. 140826), the Graduate School of Aalto ELEC, and the Aalto Media Factory. We acknowledge the work of Inger Ekman and Koray Tahiroğlu in the first evaluation and visualization of iPalmas, respectively. We also thank Antti Ikonen and Ferhat Şen for new visualization ideas and the evaluation participants for giving us a hand (or two) in exploring rhythmic interaction.

7. REFERENCES

[1] C. Chafe and M. Gurevich. Network time delay and ensemble accuracy: Effects of latency, asymmetry. *Proceedings of the AES 117th Convention*, Oct. 2004.

[2] A. Jylhä, I. Ekman, C. Erkut, and K. Tahiroglu. Design and evaluation of human-computer rhythmic interaction in a tutoring system. *Computer Music J.*, 2011. Accepted for publication.

[3] C. Larman. *Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Prentice Hall PTR, October 2004.

[4] Z. Obrenovic, J. Abascal, and D. Starcevic. Universal accessibility as a multimodal design issue. *Communications of the ACM*, 50(5):83–88, 2007.

[5] Z. Obrenovic and D. Starcevic. Modeling multimodal human-computer interaction. *IEEE Computer*, 37(9):65–72, 2004.

[6] S. O'Modhrain. A Framework for the Evaluation of Digital Musical Instruments. *Computer Music J.*, 35(1):28–42, 2011.

[7] A. Sears and J. Jacko. *The Human-Computer Interaction Handbook*. Fundamentals, evolving technologies, and emerging applications. Lawrence Erlbaum Associates, 2nd edition, 2008.

A hair ribbon deflection model for low-intrusiveness measurement of bow force in violin performance

Marco Marchini[†]
marco.marchini@upf.edu

Panos Papiotis[†]
panos.papiotis@upf.edu

Alfonso Pérez[†]
alfonso.perez@upf.edu

Esteban Maestre[†‡]
esteban.maestre@upf.edu
esteban@ccrma.stanford.edu

[†] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[‡] Center for Computer Research in Music and Acoustics, Stanford University, USA

ABSTRACT

This paper introduces and evaluates a novel methodology for the estimation of bow pressing force in violin performance, aiming at a reduced intrusiveness while maintaining high accuracy. The technique is based on using a simplified physical model of the hair ribbon deflection, and feeding this model solely with position and orientation measurements of the bow and violin spatial coordinates. The physical model is both calibrated and evaluated using real force data acquired by means of a load cell.

Keywords

bow pressing force, bow force, pressing force, force, violin playing, bow simplified physical model, 6DOF, hair ribbon ends, string ends

1. INTRODUCTION

Violin is regarded as among the most complex musical instruments, making different control parameters available for the performer to freely shape rich timbre characteristics of produced sound. Within the different bowing control parameters, only the bow transversal velocity could be considered as of comparable importance as the bow pressing force exerted by the player on the string [2]. When approaching the study of violin performance from a computational perspective, the accurate acquisition of control parameter signals appears as highly desirable, as it has been demonstrated by the research effort devoted to such pursuit during the past few years [1, 12, 7, 3, 5, 8]. In particular, the measurement of bow pressing force not only has received special attention because of its key role in timbre control, but also because of a number of measurement-specific issues that appear as harder to overcome, as it is accuracy, robustness, or intrusiveness.

An early attempt to pursue the measurement of bow pressing force from real violin practice dates back to 1986. Askenfelt [1] used wired strain gages at the frog and the tip in order to infer the bow pressing force applied on the string. Although useful for the instrument-modeling purposes of the authors, significant intrusiveness would make difficult the use of such a system in a real performance scenario.

Intrusiveness improved significantly by a first wireless acquisition system proposed by Paradiso [8], who attached a resistive strip to the bow which was driven by an antenna mounted behind the bridge of the cello. A measurement relative to the bow pressing force was carried out by using a force-sensitive resistor below the forefinger. Despite the reduction on intrusiveness, the obtained measure resulted rather unrelated to the actual force exerted on the string, as it happened to Young's approach [12], who measured downward and lateral bow pressure with foil strain gages permanently mounted around the midpoint of the bow stick.

The first effort to relate the strain of the bow hair as a measure of force was carried out by Rasamimanana [9], although the technique reached its first state of maturity (in terms of accuracy) with the technique introduced by Demoucron [4] and more recently reused and improved by Guaus [5]: the deflection of the hair ribbon is measured at the frog (and also at the tip in one of the earlier versions) by using a strain gage attached to a plate laying against the hair ribbon which bends when the string is pressed. This technique, while providing surprisingly good estimations of bow force, suffers from remarkable intrusiveness and reduced robustness, making difficult its prolonged use in stage or performance contexts.

In this paper, we present a methodology for the estimation of bow pressing force by using a simplified physical model of the hair ribbon deflection which makes use of only position and orientation (6DOF) measurements on the bow and violin. The motivation is to minimize the intrusiveness by avoiding the use of additional sensors, and therefore construct a more reliable system that can be used more naturally for longer periods of time. The principal source of information comes from measuring, as it was already proposed by Maestre [7], the distance between the ideal (no deflection) segment defined by the ends of the hair ribbon, and the segment defined by the ends of the string being played. A physical model of the hair ribbon deflection is constructed and calibrated from real data measurements using a load cell, and used later for estimating the bow force in real performances.

The rest of the paper remains as follows. Section 2 introduces the measurement system and outlines the features used in our study. In Section 3 we present a simplified physical model for a single hair thread and then generalize it to describe the complete hair ribbon. Section 4 describes a procedure to minimize the deviation between our model and recorded force data. We conclude with some prelimi-

NIME'11, 30 May–1 June 2011, Oslo, Norway.

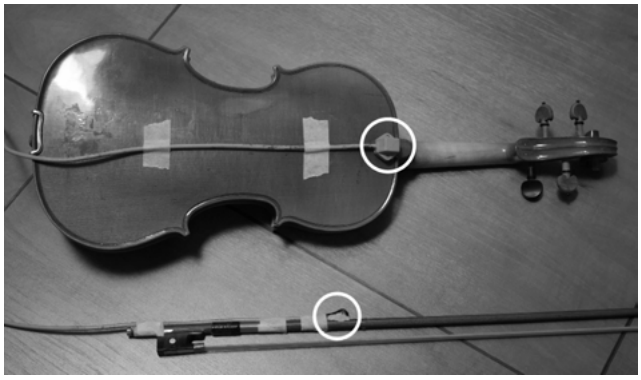


Figure 1: Detail of violin and bow placement of sensors during a performance recording.

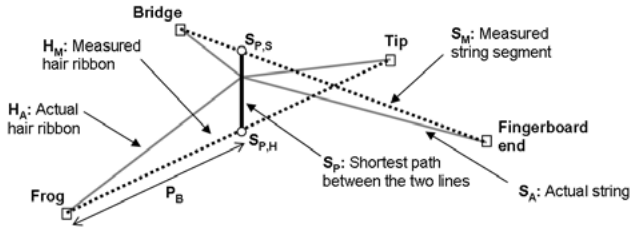


Figure 2: Measured string and hair ribbon segments, computed from their extracted end points, versus their actual configuration. Deformations have been exaggerated in order to illustrate the importance of segment S_P .

nary results along with future directions.

2. MEASUREMENT SYSTEM

To obtain the physical parameters necessary for the estimation of vertical bow force against the violin string, we utilize the methodology described in [5, 7]. Essentially, the methodology consists in (i) acquiring instrumental gesture parameters (such as bow transversal position and hair ribbon-to-violin string distance), and (ii) using a load cell to measure applied force during calibration and evaluation.

2.1 Acquisition of violin instrumental gesture parameters

The acquisition of the instrumental gesture parameters is done in real-time using the Polhemus Liberty system[1], a 6DOF tracking system based on electromagnetic field sensing (EMF), and consisting of two wired miniature sensors and a transmitting source. Each sensor provides three 3DOF for position and 3DOF for orientation, both at 240Hz sampling rate, with static accuracies of 0.75mm and 0.15 degrees RMS respectively. These sensors are respectively placed on the bow and the body of the violin, as seen in Figure 1. From the position and orientation data provided by these two sensors, and thanks to a calibration procedure involving a third sensor, we are able to obtain the position of the ends of the strings, and the (ideal, assuming no deformation) position of of the four ends of the hair ribbon (considered as having finite width), as detailed in [7].

Having obtained the position of the strings as well as the bow hair ribbon, we can proceed to calculate several parameters regarding the position and orientation of the bow with respect to the violin strings (see Figure 2). Particularly for this model, the most relevant parameters are:

1. **bow transversal position**, also referred to as *bow displacement*; this is is computed as the euclidean distance between $S_{P,H}$ and the measured frog end of the hair ribbon.

2. **bow-string distance**, also referred to as *pseudo-force*; the modulus of the intersecting line segment S_P , which is perpendicular to the string and to the hair ribbon. This segment will get longer for higher deformations due to pressing force. Thus, we compute a euclidean distance between $S_{P,S}$ and $S_{P,H}$ between each of the two longitudinal edges (left, right) of the hair ribbon (tracked as a finite surface instead of as it is shown in Figure 2 for the sake of simplicity).

[1] www.polhemus.com

2.2 Measurement of applied force

In order to both design and evaluate our system, we used a linear load cell to measure the actual force being applied by the bow, as suggested by [10] and implemented in [5]. The cell is fixed to a wooden support, while a thin methacrylate cylinder is placed over the cell to simulate a virtual string. By using a similar calibration method described above, we are able to track the ends of the cylinder and thus acquire a number of bowing parameters (including *bow displacement* and *pseudoforce* as simultaneously recorded along with the output of the load cell.

The output of the linear load cell itself is calibrated using a set of precision weights; the force produced by these weights on the load cell is derived from Newton's second law of motion, $F = Mg$, with $g = 9.8m/s^2$. The voltage output of the load cell is post-processed to match the corresponding unit of Newtons by applying a simple linear transformation of the form $y = qx + s$, where q equals the voltage gain and s equals the voltage offset.

3. A SIMPLIFIED PHYSICAL MODEL FOR HAIR RIBBON DEFLECTION

In this section we present a simplified physical model of a flexible thread or hair as appearing in a violin bow, and then we extend it to the case of multiple hairs and generalize it to describe the complete hair ribbon. We use such physical model in order to approximate, given solely information extracted from 6DOF sensors, the force exerted on the string regardless of the displacement or tilting of the bow. An important simplification was to assume the bow stick as rigid.

3.1 The thread

The simplest approximation of the bow hair-ribbon is a single elastic thread stretched between two points A and B (see Figure 3). At its rest position the thread has a length of l, which coincides with the distance between the points A and B. When a force is applied on a point C, the thread stretches and is elongated until an internal equilibrium of the system is reached.

In its rest position, we consider such thread as the limit of an array of masses connected by springs, presenting a mass-to-mass distance approaching zero. We parameterize the thread by a function $u : [0, 1] \to \mathbb{R}^2$, and express the potential energy of the thread as

$$\frac{1}{2}\frac{T}{l}\int_0^1 u'(t)^2 dt, \qquad (1)$$

where T is the tension of the thread. If u is the parametriza-

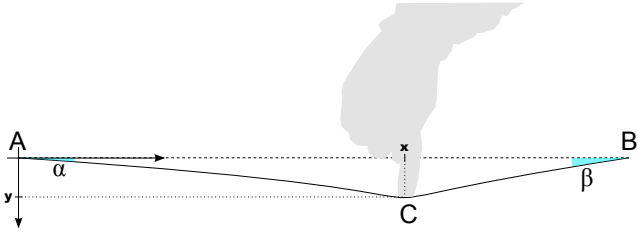


Figure 3: A single elastic thread fixed at two extremes A and B is stretched by applying a force in a point C.

tion of the thread of Figure 3 with $u(0) = A$, $u(c) = C$, and $u(1) = B$ where $c \in]0, 1[$, the potential energy is given by

$$\frac{1}{2}\frac{T}{l}\left(\frac{x^2+y^2}{c} + \frac{(1-x)^2+y^2}{1-c}\right). \tag{2}$$

The internal equilibrium, i.e. the minimum potential energy, is reached for $c = c_{\text{eq}}$, where

$$c_{\text{eq}} = \frac{x^2+y^2}{x^2+y^2+\sqrt{((l-x)^2+y^2)\,(x^2+y^2)}}. \tag{3}$$

In this equilibrium state, the point C is subject to two forces $\overrightarrow{f_1}$ and $\overrightarrow{f_2}$ in the direction CA and CB respectively. Their magnitude is the following:

$$\|\overrightarrow{f_1}\| = T\,(l_1 - c_{\text{eq}} l) \tag{4}$$

$$\|\overrightarrow{f_2}\| = T\,(l_2 - (1 - c_{\text{eq}})\,l) \tag{5}$$

Let's denote l_1 and l_2 the length of the AC and CB parts respectively, α and β the angles CAB and ABC respectively, and $\Delta l = l_1 + l_2 - l$ the total en-lengthening of the thread. The total force that the thread exerts on C is $\overrightarrow{F} = \overrightarrow{f_1} + \overrightarrow{f_2}$. If we set a coordinate system at the center of the thread as shown in Figure 4, the point C will be described by its coordinates (x, y). Now, writing $\overrightarrow{F} = (F_{\text{horz}}, F_{\text{vert}})$ where F_{horz} is the horizontal component of $\overrightarrow{F}$ and F_{vert} the vertical component, the vertical component of the force can be considered as the force applied to the string, and written (observing that $\sin(\alpha) = \frac{y}{l_1}$ and $\sin(\beta) = \frac{y}{l_2}$) as

$$F_{\text{vert}}(x, y) = \|\overrightarrow{f_1}\|\frac{y}{l_1} + \|\overrightarrow{f_2}\|\frac{y}{l_2}. \tag{6}$$

3.2 The Hair Ribbon

A more precise approximation of the hair ribbon is to consider it as a a strip of parallel threads, assuming that the force exerted by the ribbon is the sum of the contributions of each thread. Considering an homogeneous distribution of threads determined by a constant ρ and if we define w to be the width of the strip, we can define the force applied to the string as

$$\text{Force} = \rho \int_0^w f(z)dz, \tag{7}$$

where $f(z)$ is the force density of the thread situated at position z on the strip.

Let $m := y(0)$ and $M := y(w)$ be respectively the measured left-hand side and right-hand side *bow-string distance* (see Section 2.1). We then have

$$y = m + \frac{(M-m)}{w}z. \tag{8}$$

Figure 5 schematically depicts the possible relative positions (displacements) that we considered for the hair ribbon

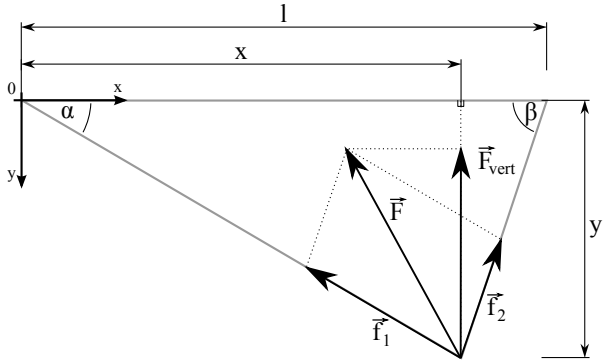


Figure 4: Schematics of the single thread as in Figure 3. The chosen coordinate system is shown in the upper left corner. Note that the y-axis increases towards the bottom. The reaction forces of the threads are drawn. The angles α and β have been exaggerated in the picture so that also the length y and all the force components could be clearly visible.

(transversal view) as relative to the string. The displacement of the string with respect to the bow is determined by the linear relation 8. Only the threads where y is positive are contributing to the force (as they are in contact with the string). This happens[2] when $z > -\frac{wm}{M-m} =: \psi$. Having the diagram of Figure 4 as a reference, we considered the variable x as constant with respect to z while, depending on the changes of y, we reduce the problem to three main cases, defined as

Case I : The y are negative (the ribbon is not touching) with respect to all $z \in [0, w]$, having

$$f(z) = 0\ \forall z; \tag{9}$$

Case II : y is positive for $0 < z < \psi < w$ reaches 0 for $z = \psi$ and is negative for $z > \psi$, with

$$f(z) = \begin{cases} 0 & \text{for } z \in [0, \psi] \\ F(x, y(z)) & \text{for } z \in (\psi, w] \end{cases}; \tag{10}$$

Case III : y is positive for all $z \in [0, w]$, so

$$f(z) = F(x, y(z)). \tag{11}$$

Case I is of little significance, since the force is zero. In the other two cases, applying equation (8) in the equations (10) and (11), we may completely rewrite equation (7) using the definition of f and Δl canceling thus all the indirect dependencies to the variable y. Applying then the substitution $z = \frac{w}{M-m}(y - m)$ to the integral we can write the results in term of the function

$$\tilde{F}(z) := \frac{1}{T}\int_0^z F_{\text{vert}}(x, y)dy, \tag{12}$$

where we divide by T so that $\tilde{F}$ do not depend on the tension. We will handle this parameter in the further formulas.

For the fundamental theorem of calculus plus taking into account the term $\frac{w}{M-m}$ of the substitution, we conclude, for the three considered cases, as

Case I : The y are negative (the ribbon is not touching) with respect to all $z \in [0, w]$, thus

$$\text{Force} = 0;$$

[2]Considering, for the moment, the case where $M > m$ without any loss of generality.

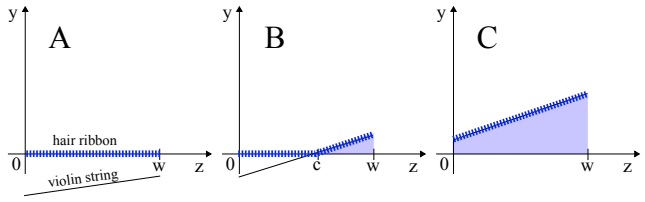


Figure 5: I. Non touching ribbon. II. Partially touching. III. Fully touching.

Case II : y is positive for $0 < z < \psi < w$ reaches 0 for $z = \psi$ and is negative for $z > \psi$, having

$$\text{Force} = T\rho w \frac{\tilde{F}(\max(M,m)) - \tilde{F}(0)}{|M-m|}; \qquad (13)$$

Case III : y is positive for all $z \in [0, w]$, so

$$\text{Force} = T\rho w \frac{\tilde{F}(M) - \tilde{F}(m)}{M-m}. \qquad (14)$$

Finally, we observe that:

1. The final force **only** depends on the variables x, M and m plus the constants T, ρ and w.

2. The cases I, II and III can be identified only looking at M and m. Case I holds when both are negative, case II when they differ in sign and case III when both are positive.

3. Considering $M > m$ did not cause a loss of generality. Indeed, for $M < m$, due to a symmetry of the problem, we could just switch M with m but this, thanks to the way equation (13) has been expressed, does not change the result. Finally we can interpret the case $M = m$ as a limit case of equation (14) when $(M - m) \to 0$. The results of the limit is, in fact, $\text{Force} = \rho w F_{\text{vert}}(x, M)$ corresponding to an equal contribution of all the threads to the final force.

4. OPTIMIZATION PROCEDURE

The described model is parametrized by a single scalar value given by the product $T\rho w$. Changing this parameter will affect the whole prediction, scaling it by a factor. Thus, our initial idea was to infer this factor from an experiment; however, there are additional conditions in the real case which are not addressed by the physical model. First, the motion tracking sensor placed on the bow stick might rotate by a small angle θ after the calibration has been performed, causing a rotation of all the data. Secondly, due to the movement of the sensor, it might be necessary to adjust the offsets of the bow displacement adding a constant a, and the offset of the vertical distance with a constant b. Finally, in order to address the problem of the bending of the stick, another constant r is added defining a transformation which will compensate, the effect of the stick bending by dividing the pseudoforce by a value depending on the bow displacement. The final transformation is given by the following formula:

$$\begin{cases} x' = & a + x \\ M' = & Map_{(x,b,\theta,r)}(M) \\ m' = & Map_{(x,b,\theta,r)}(m) \end{cases}, \qquad (15)$$

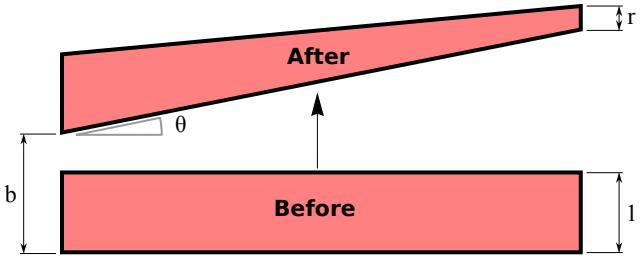


Figure 6: The function Map applied to a rectangle. For this example exaggerated parameters used are $r = 0.5$, $b = 1.5$, $\theta = 0.1$rad.

where

$$\text{Map}_{(x,b,\theta,r)}(y) := (b + y\left(\frac{1+r}{2} + \frac{(-1+r)\left(-\frac{l}{2}+x\right)}{l}\right)\text{Cos}[\theta] + x\text{Sin}[\theta]) \qquad (16)$$

In Figure 6 it is illustrated how the defined transformation alters a rectangle with some fixed inflated parameters.

4.1 Description

Suppose we have a training set $\{(x_i, M_i, m_i)\}_{i=1,\dots,n}$ where x_i is the bow displacement, M_i is the pseudo-force of left side and m_i is the pseudo-force of right side at time i. Given the parameters T, θ, a, b and r we consider the prediction

$$\text{Force}_i(T,\theta,a,b,r) = \text{Force}(x_i + a, \text{Map}_{(x_i,b,\theta,r)}(M_i), \text{Map}_{(x_i,b,\theta,r)}(m_i)). \qquad (17)$$

We want to find the better values for the parameters in order to minimize the absolute error:

$$J(T,\theta,a,b,r) = \frac{1}{2}\sum_{i=1}^{n} (\text{Force}_i(T,\theta,a,b,r) - \text{nidaq}_i)^2. \qquad (18)$$

We thus aim at finding:

$$(T^*,\theta^*,a^*,b^*,r^*) = \underset{(T,\theta,a,b,r)}{\arg\min}\ J(T,\theta,a,b,r) \qquad (19)$$

We use the Nelder-Mead simplex method [6] in order to find a local minimum, starting from the identity transformation parameters: $T = r = 1$, $\theta = a = b = 0$. In order to reduce the computation time, we down-sampled the signal to 8 samples a second. Before the optimization of the parameters we also filtered the dataset, to remove noisy data. We removed the samples where the measurement of the Force cell was less than 0.2. In fact the sensitivity of the sensor for small forces is reduced and noisy.

4.2 Results

Using the acquired gesture parameters along with the Force cell data, we recorded three evaluation datasets. In the *dataset 1*, an almost constant force was applied with different bow transversal positions and different tilts. In the *dataset 2*, the pseudo-force was changing constantly from positive to negative while changing tilt and bow transversal position in order to simulate the way violin is normally played. In the *dataset 3*, bow transversal positions was kept fixed while the force and the tilt where changing. This was done for many different bow transversal positions. Each recording was around one minutes long. We created an *Joint Dataset*, with the samples of the three recordings and

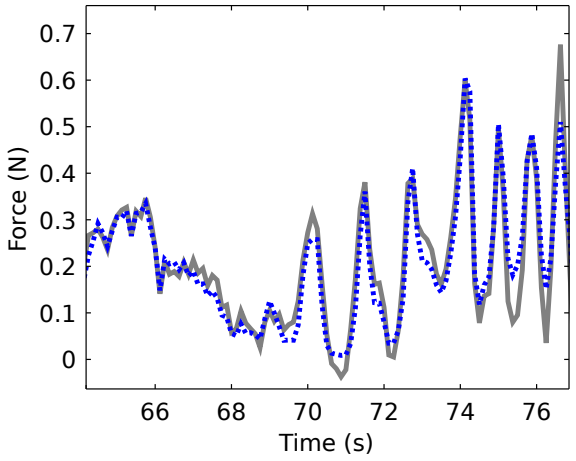


Figure 7: An excerpt of the recorded force signal *(continuous gray line)* along with its prediction *(blue dashed line)* after the optimization has been performed.

we performed a 10-fold cross validation, with a performance going to 13.75 relative error.

Finally, in order to investigate on the best calibration procedure, we also compared the three different datasets. The table 1 shows the results of each possible combination of training-set with test-set. Is clearly shows that even though the second dataset performs best to predict the rest (with 97.4 mean correlation) the other two datasets give comparable results. So even though trying to replicate the variability of parameters of a real performance in the calibration lead to slightly better results, even a short calibration of one minute, with static bow displacement positions results to be good for a general purpose force estimation.

5. CONCLUSION

We presented a physical model of the violin bow assuming a rigid bow stick. We estimated the force of the bow on the string for each configuration of the hairs considering the bow transversal position and the bow-string-distance at the left and right sides of the hair ribbon. We sampled some datasets from with the Polhemus equipment encountering systematic parallax-like errors in the data caused by human error in the calibration procedure or by the sensors. We defined a transformation to correct those effect dependent on 4 parameters. We thus used the training datasets to fit the transformation parameters plus the the tension T of the hairs.

The bow model gives an estimation of the bowing force in Newtons with a very high correlation coefficient for the half of the bow near to the frog. Additionally, by comparing different datasets corresponding to different types of bowing, we are able to identify the type of data that is sufficient for obtaining a good prediction. This way we found out a procedure to calibrate the model in a few seconds.

The main application of this bow physical model is to complement a sensing system for the acquisition of bowing gestures by providing accurate measurements of the force that the bow is exerting on the string while allowing for less intrusive capturing devices. Additionally, the model can be used to build or improve data acquisition for sound controlling interfaces.

Table 1: Relative error and Correlation of the prediction with the true force for each training set and test set coupling.

		Test Set							
		Dataset 1		Dataset 2		Dataset 3		Joint	
Training Set	Dataset 1	corr	rel	corr	rel	corr	rel	corr	rel
		89.21	16.75	96.18	23.68	98.84	14.67	97.29	17.03
	Dataset 2	corr	rel	corr	rel	corr	rel	corr	rel
		95.03	13.78	97.76	16.75	98.37	20.13	97.4	16.88
	Dataset 3	corr	rel	corr	rel	corr	rel	corr	rel
		84.95	23.61	95.87	22.64	98.88	12.18	96.13	19.82
	Joint	corr	rel	corr	rel	corr	rel	corr	rel
		94.1	13.04	97.3	19.73	98.94	12.19	98.12	13.75

6. FUTURE WORK

A clear potential improvement that will be carried out in the future is the estimation of the stick bending effect. By looking at histograms of bow displacement in real playing, we observed that musicians use the lower part of the bow (closer to the frog) significantly more often, which reduces the stick bending effect. When evaluating our model for extreme cases in which the majority of the frames were recorded when the performer was playing near the tip, the performance gets significantly reduced. This could be explained by the fact that in our model we did not address explicitly the effect of the stick bending. The effect is, actually, too big there to be corrected from the mapping in that region of the bow. A further step will be to include the deflection of the string in the model. Such a study should lead to a complete force estimation system for all the parts of the bow while providing a completed physical model of the bow. Such a model will, of course, improve obtained results. However, because of the non-linearity of the model, the precision of the prediction varies according to the bow displacement. This has to be considered an intrinsic limitation of any deflection model of the bow and, as suggested by a reviewer, a thorough study on the propagation of noise in the formulas should be carried out in the future.

Regarding the parallax error arising from the Polhemus equipment, it would be interesting to reproduce the experiment with other type of measurement such as IR camera-based MOCAP (Qualysis) for a comparison of the prediction error.

7. ACKNOWLEDGEMENTS

This work was supported by the EU FP7 FET SIEMPRE Project.

8. REFERENCES

[1] A. Askenfelt. Measurement of bow motion and bow force in violin playing. *The Journal of the Acoustical Society of America*, 80(4):1007–1015, October 1986.

[2] L. Cremer. *Physics of the Violin*. The MIT Press, Cambridge, Massachusetts, USA, November 1984.

[3] M. Demoucron, A. Askenfelt, and R. Causse. Measuring bow force in bowed string performance: Theory and implementation of a bow force sensor. *Acta Acustica united with Acustica*, 95(4):718–732, 2009.

[4] M. Demoucron and R. Caussé. Sound synthesis of bowed string instruments using a gesture based control of a physical model. In *Proceedings of the 2007 International Symposium on Musical Acoustics*, Barcelona, 2007.

[5] E. Guaus, J. Bonada, E. Maestre, A. Perez, and

M. Blaauw. Calibration method to measure accurate bow force for real violin performances. In G. Scavone, editor, *International Computer Music Conference*, pages 251–254, Montreal, Canada, 16/08/2009 2009. The International Computer Music Assiciation, The International Computer Music Assiciation.

[6] J. Lagarias, J. Reeds, M. Wright, and P. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1999.

[7] E. Maestre, J. Bonada, M. Blaauw, A. Perez, and E. Guaus. Acquisition of violin instrumental gestures using a commercial emf device. In *International Computer Music Conference*, Copenhagen, Denmark, 27/08/2007 2007.

[8] J. A. Paradiso and N. A. Gershenfeld. Musical applications of electric field sensing. *Computer Music Journal*, 21(2):69–89, 1997.

[9] N. Rasamimanana. Gesture analysis of bow strokes using an augmented violin. Master's thesis, IRCAM, Paris, Fance, 2003.

[10] E. Schoonderwaldt. *Mechanics and acoustics of violin bowing*. PhD thesis, Stockholm Royal Institute of Technology, Stockholm, Sweden, 2009.

[11] E. Schoonderwaldt and M. Demoucron. Extraction of bowing parameters from violin performance combining motion capture and sensors. *The Journal of the Acoustical Society of America*, 126(5):2695–2708, 2009.

[12] D. S. Young. Wireless sensor system for meassurement of violin bowing parameters. In *Proceedings of the Stockholm Music Acoustics Conference*, Stockholm, Sweden, 2003.

Random Access Remixing on the iPad

Jon Forsyth, Aron Glennon, Juan P. Bello
Music and Audio Research Lab (MARL)
New York University, New York, NY USA
{jpf211, apg250, jpbello}@nyu.edu

ABSTRACT

Remixing audio samples is a common technique for the creation of electronic music, and there are a wide variety of tools available to edit, process, and recombine pre-recorded audio into new compositions. However, all of these tools conceive of the timeline of the pre-recorded audio and the playback timeline as identical. In this paper, we introduce a dual time axis representation in which these two timelines are described explicitly. We also discuss the random access remix application for the iPad, an audio sample editor based on this representation. We describe an initial user study with 15 high school students that indicates that the random access remix application has the potential to develop into a useful and interesting tool for composers and performers of electronic music.

Keywords

interactive systems, sample editor, remix, iPad, multi-touch

1. INTRODUCTION

The remixing and processing of audio samples[1] is a common feature in the creation and production of popular music, especially electronic music. Artists use a multiplicity of hardware and software tools to edit, process, and recombine prerecorded audio to create entirely new compositions. Many of these tools operate under the "mixing console" paradigm, wherein audio is recorded onto separate audio tracks, usually corresponding to a mixer channel. Most software tools feature waveform editing capabilities, and allow for a variety of processes, e.g. filtering or other audio effects, to be applied to tracks, either individually or in groups.

However, all these applications operate on a single playback timeline in which audio samples are placed. Such an approach fails to explicitly represent the timeline of the original samples. In this paper we introduce a *dual time axis* representation in which the playback and the sample timelines are placed on separate axes in a two-dimensional space, and present a *random access remix* application for the iPad, an implementation based on this representation. We argue that the dual time axis representation can provide users with an intuitive interface for the temporal restructuring of audio samples, and can therefore support the development of novel tools for the composition and performance of electronic music.

The rest of this paper is organized as follows. Section 2 discusses a few commercial and non-commercial sample-editing tools currently available to users. Section 3 introduces and justifies our approach, and discusses some of its implications. In section 4 we present an iPad implementation of our design, and provide a detailed description of the user experience. Section 5 documents our initial user tests in the context of an educational activity for high school students. Section 6 concludes our presentation and discusses ideas for future work.

2. RELATED WORK

There are many commercial applications that can be used for creating remixes and loop-based compositions from prerecorded audio. Many of these applications are based on the mixing console paradigm, in which the application serves as a virtual recording studio. Typically, audio is recorded into tracks that correspond to mixer channels, processed in a variety of ways, and placed at arbitrary points along a common playback timeline. Some examples include commercial applications, such as Logic, GarageBand, and ProTools, and free- and share-ware software such as Audacity[2]. These applications often feature a waveform editor, which allows the user to directly edit an audio sample, although the editor is generally not central to the user interface. Other commercial applications are designed more specifically for the task of waveform editing. For example, Propellerhead's ReCycle[3] is a devoted sample editor that segments a sample, and allows the user to modify each segment independently, e.g., by altering its start and end points and by applying audio processing.

The academic community has produced a number of sample editing tools. For example, the waveTable [6] is a sample editor in which the waveform is displayed on a multi-touch tabletop interface. Touch gestures serve as navigation commands, allowing the user to scroll through the waveform and zoom in or out. A set of tangible tools allows the user to edit the sample and add various audio effects. The Slidepipe [1] is a physical controller consisting of a number of horizontal pipes, each with a set of paddles and ropes, that are used to control audio effect parameters settings and the start and end points of a sample. A similar system, the Chopping Board [4], uses a touch sensitive pad as the physical interface. An audio sample is mapped to the pad, with the location at which the user touches the pad indicating the start point of the sample. Two faders and a knob are used to control effects settings and select samples.

[1]The term sample is used here in the colloquial sense; i.e., a segment of audio.

[2]`http://audacity.sourceforge.net`

[3]`http://www.propellerheads.se/products/recycle`

NIME'11, 30 May–1 June 2011, Oslo, Norway.

In addition, there are a number of loop-based sample remix systems. The Beat-Sync-Mash-Coder [3] is a web-based system in which users can combine a number of pre-recorded audio loops into "mash-ups." The user can also alter the overall tempo. The music loop explorer system [7] also allows a user to mix together pre-recorded samples to create mash-ups. In this case, the system automatically segments music tracks and computes a similarity measure between the segments. The user can combine different loops together and set a master tempo, with the system automatically adjusting the tempo of each segment. Unlike the systems discussed above, the Beat-Sync-Mash-Coder and the music loop explorer system treat a sample as an atomic entity; i.e., it cannot be edited, only combined with other samples and globally modified.

3. APPROACH

3.1 Dual Time Axis Representation

The systems discussed above allow the user to temporally restructure audio, for instance by segmenting the sample and then rearranging the resulting segments, or to combine different audio samples. However, all of these systems treat the audio sample timeline and the playback timeline as collinear; in other words, these two timelines are identical. While this conception of time is appropriate for certain tasks, it also obscures the fact that the sample and playback timelines are distinct. A paradigm that makes this distinction explicit could facilitate new and interesting manipulations of an audio sample.

Perhaps the most straightforward of these manipulations is the temporal restructuring of an audio sample. We want to have the ability to place any region of a sample at any point in the playback timeline. This requires precise control over both the start and end times of each audio event, as well as the start and end times of the desired region from the source audio sample. Both of these sets of parameters are temporal, and both represent positions on separate timelines: the first set of parameters specifies a location on the playback timeline, and the second specifies a location on the source audio timeline.

The above observations suggest a representation that uses two time axes. One such representation is the *recurrence plot*, a tool used to analyze and visualize nonlinear dynamic systems [5]. A standard recurrence plot is a two dimensional, square, binary matrix generated by comparing the state of a dynamic system at a particular time with the states of the system at all times. If a state at time i is the same (within a margin of error) as another state, at time j, a value of 1 is entered into the matrix at position (i, j); otherwise, the value at (i, j) is set to 0. Thus, the row and column indices of the matrix represent time.

Using the recurrence plot as inspiration, we developed a representation consisting of two temporal axes, which we refer to as the *dual time axis* representation. Unlike the standard recurrence plot, in which both dimensions represent a position along the same timeline, here the horizontal axis describes playback time, and the vertical axis describes the timeline of the source sample. This representation allows *random access* to the source audio sample. That is, any portion of the sample can be played back at any position along the playback timeline.

3.2 Interactions

Figures 1 and 2 depict the dual time axis representation. In these figures, the waveforms of the source audio and the output audio are shown to illustrate the interactions. Time 0 for both axes is in the upper left corner. The basic interactions available to the user are the ability to draw, resize, and move blocks. In order to specify a particular region of audio from the source sample to playback, the user draws a block, as shown in Figure 1. The start and end times of the desired region of audio from the source sample are specified by the vertical coordinates of the block, and the start and end times at which this region of audio is played back are specified by the horizontal coordinates. For example, in Figure 1, if the user draws the block as shown, the region of the source sample between 1.0 sec and 2.0 sec is played back starting at time 2.5 sec.

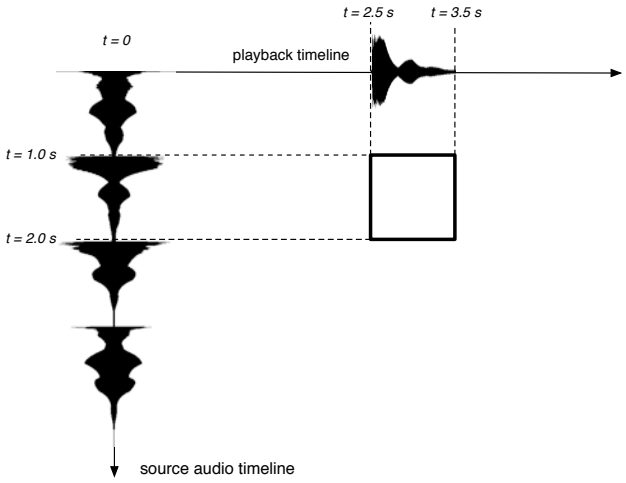


Figure 1: Dual time axis representation.

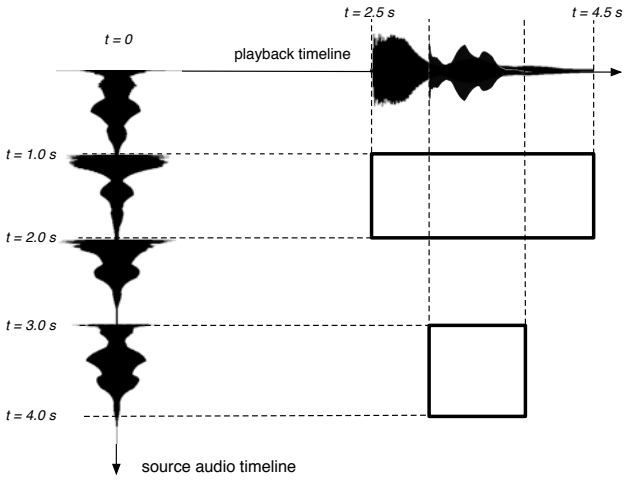


Figure 2: Block in figure 1 time stretched, with additional block.

The user is not restricted to drawing square blocks, but can either draw a rectangular block or resize an existing square block into a rectangle. In the case of a rectangular block, the length of the region from the source audio will not equal the length along the playback timeline. Thus, we must either time- stretch or time-compress the specified region of source audio so that it fits within the specified playback time. Figure 2 shows the effect of resizing the block from Figure 1 into a rectangle. As in Figure 1, the region of source audio is between 1.0 sec and 2.0 sec, a duration of 1 sec. However, the rectangle indicates that this region of audio should be played back between 2.5 sec and 4.5 sec, a duration of 2 sec. Therefore, the region of source audio is time-stretched by a factor of 2.

We can think of each block as having a local copy of the

region of audio defined by its vertical coordinates. Therefore, when we modify the content of a block, for instance by time-stretching or time-compressing it, we are altering only the local copy, while the original source sample is left unaffected. As a consequence, any modifications to a particular block's audio leaves the others' audio unaffected, allowing us to apply any type of audio processing independently. The user can also draw multiple blocks, as shown in Figure 2; here, another block has been added to the configuration shown in Figure 1. The resulting output is a sum of each block's audio.

We refer to an application based on the representation of time described above as a *random access remix application*. It is our contention that such an application would allow a user to remix an audio file with precision equal to that available in typical remix or sample editors. Further, we believe that the dual time axis representation will encourage the user to view an audio sample as random access data instead of data that must be accessed sequentially, and that this new perspective will inspire new ways of thinking about sample editing and remixing.

4. IMPLEMENTATION

4.1 Technical Details

We chose to implement the random access remix application on an Apple iPad, a multi-touch device with a rich development environment (APIs, libraries, frameworks, etc.), including native support for a wide variety of gestures. The relatively large screen and processing power of the iPad also make it an excellent platform on which to develop a musical application with an intuitive and natural graphical user interface (GUI). However, the basic design does not require a touchscreen, and could be implemented on a desktop or laptop computer.

We implemented a prototype application in Objective-C and C/C++, using a combination of Apple's UIKit framework and OpenGL ES to implement the GUI. OpenGL ES is an implementation of OpenGL for mobile devices, including the iPad. It is efficient, powerful, and portable. In addition, we used MoMu, an open source application development toolkit for mobile devices [2]. MoMu consists of APIs and utilities that support the development of interactive mobile applications on iOS, including a layer of abstraction that greatly simplified the handling of audio input and output. We subsequently developed a second version of the application, using OpenGL ES, MoMu, and Cocos2D[4], a free, open source framework for developing graphical applications for the iPhone and iPad. Cocos2D provides functionality similar to the UIKit, but with greater flexibility.

4.2 User Interface

The main user interface for the second version of the random access remix application is shown in Figure 3. The interface consists of four main elements: the file select/effects edit region (① in Figure 3), the waveform display (②), the block editing region (③), and the toolbar (④). To load a sample, the user selects a file listed in the file select region. Upon doing so, the audio waveform is displayed in the waveform display region, with time 0 at the top. The block editing region is the area in which the user creates and modifies blocks. As in Figures 1 and 2, the horizontal axis represents the playback timeline (with time increasing from left to right), and the vertical axis represents the sample timeline (with time increasing from top to bottom).

The toolbar (③ in Figure 3) allows the user to control playback, clear the block editing region, copy or delete a

[4]http://www.cocos2d-iphone.org/

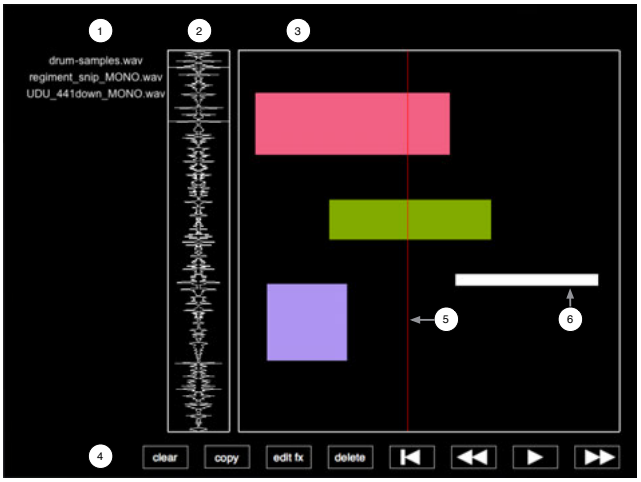


Figure 3: User interface.

block, and edit the effects settings for a block. The playback controls consist of the standard set of tape deck controls: reset playback position to time 0, rewind, play/pause, and fast forward. The audio output is looped, so that when playback reaches the end of its timeline (the rightmost edge of the block editing region), it immediately returns to time 0 (the leftmost edge of the block editing region). The vertical red line in the block editing region (⑤ in Figure 3) indicates the current playback position, sweeping from left to right during playback.

The user can edit the effects processing applied to a particular block by pressing the "edit fx" button in the toolbar (④ in Figure 3). Pressing this button reveals a set of sliders, used to adjust effects parameters, in the file select/effects edit region (①). The list of available files is displayed if the user presses this button again.

The block editing region responds to standard iOS gestures. To draw a block, the user places a finger in an empty region of the block editing region, dragging until the block is the desired size. In order to select an existing block, the user places a finger on the block; the block color changes to white to indicate selection. Figure 3 shows a selected block (⑥). Once selected, the user can copy, delete, or set the effects parameters applied to the block using the corresponding buttons in the toolbar. In addition, the user can move a block by dragging it to the desired location.

5. INITIAL EXPERIENCES

5.1 Context

The first implementation of the random access remix application[5] was used as part of a four session long workshop for high school students developed by New York University and the Institute for Collaborative Education. The general objective of the workshop was to use music as a basis for teaching the high school students various scientific concepts. Each session, the students attended a lecture, and then performed a hands-on activity related the lecture material. The workshop participants consisted of 15 students total, all in either the 10^{th} or 11^{th} grade. The group was split roughly evenly between males and females, and between students with an interest in science and those with experience in music.

The final session focused on digital audio, and included

[5]This implementation is similar to the one described in Section 4, with the most notable difference being the lack of a waveform display region.

a discussion of quantization, sampling, and the ease with which digital audio can be manipulated. Because the random access remix application is essentially a tool for the temporal restructuring of audio, its functionality aligned well with the lecture topics. In addition, we implemented a number of relevant audio effects, specifically bit crushing and downsampling. The bit crushing effect allowed the user to change the bit depth of the audio associated with a block from 16 bits to 1 bit, thus creating audible distortion at lower quantization levels. The downsampling effect allowed the user to change the sampling rate from 44100 Hz to 4410 Hz, thus raising the pitch of the audio segment as well as its overall duration. This change in duration was reflected in the block editing region of the application: lowering the sampling rate setting for a particular block reduced that block's width, thus allowing the students to both hear and see the results of downsampling a segment of audio.

5.2 Activity

We provided the students with iPads, with each iPad being shared by a group of two or three students. We briefly introduced the application to the students, describing its basic interactions, as well as the dual time axis representation. We then allowed them to freely explore the application for roughly 20 minutes, providing assistance when necessary. Although some students initially found the dual time axis representation to be confusing, they were able to understand the concept after a few minutes of explanation. Other students were comfortable with the application from the outset, and began to produce sounds soon after receiving the iPad. Most students focused on creating visual patterns of blocks and listening to the results.

When the students seemed comfortable with the application, we presented them with a musical task. We played a recording of a simple four-note target melody, and asked them to approximate it using an audio file that consisted solely of a sample of a single note of a vibraphone. In order to complete the task, the students had to complete a number of sub-tasks. First, they had to locate the beginning and end of the note in the source sample. Next, they had to place a block along the playback timeline corresponding to each note of the target melody. Finally, in order to approximate the pitches of the target melody, the students had to alter the sampling rate of each block. This task required an understanding of how digital audio can be manipulated, as well as how altering the sampling rate of digital audio alters the pitch. Most of the students were able to perform the task.

5.3 Observations

Although we were not able to obtain any quantitative data from this experience, we were able to make a number of observations. The application was, in general, positively received by the students and the teachers who accompanied them; most seemed to enjoy using it, and two or three expressed interest in obtaining a copy. However, the activity exposed a number of design flaws. In particular, during the melody creation task, the students found it difficult to properly locate the region of the audio sample corresponding to the note. This difficultly was likely due to the lack of visual feedback indicating the content of the audio sample. This problem has been remedied in the second version of the application with the waveform display, as shown in Figure 3. In addition, the students occasionally created small blocks in the interface, for instance through an accidental touch or by reducing the sampling rate to the minimum value; it was not possible to select (and thereby delete) the smallest of these blocks. Although the user could clear all the blocks, this was not a satisfactory solution. Instead, we could impose a minimum size restriction on the blocks, and increase the minimum value of the sampling rate. Although we did not implement this feature, some students inquired if was possible for the application to render the output to an audio file that they could take home.

6. CONCLUSIONS AND FURTHER WORK

The positive reaction from the high school students indicates that the dual time axis representation has merit. Although the activity with the students exposed some of the application's shortcomings, some of these problems have been fixed in the second version of the application, while others should be relatively easy to remedy. There are a number of further enhancements that we would like to make to the application, such as implementing additional audio effects. We also feel that the application should allow the user to define the length of the remix, instead of limiting it to the length of the source sample. In addition, we plan to add tools to analyze the input audio signal in order to extract high-level musical information. For instance, we feel that beat tracking and rhythmic quantization would greatly enhance the application, as it would allow the user to more easily create musical coherence between the various blocks.

While our initial experiences are encouraging, it is necessary to conduct more extensive user tests in order to produce a quantitative evaluation of the application. Such testing should identify areas of relative strength and weakness, and point the way towards further improvements. While our initial experiences indicate that our implementation is appropriate for relative novices, our hope is that an application that combines an intuitive graphical user interface with intelligent processing can achieve sufficient depth and sophistication to make it a useful tool for more experienced musicians and technologists. Further user testing could begin to provide some answers to this question.

7. REFERENCES

[1] M. Argo. The slidepipe: A timeline-based controller for real-time sample manipulation. In *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04), Hamamatsu, Japan*, 2004.

[2] N. Bryan, J. Herrera, J. Oh, and G. Wang. Momu: A mobile music toolkit. In *Proceedings of the 2010 International Conference on New Interfaces for Musical Expression (NIME2010), Sydney, Australia*, 2010.

[3] G. Griffin, Y. Kim, and D. Turnbull. Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups. In *2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Dallas, Texas*, 2010.

[4] J. Lee. The chopping board: Real-time sample editor. In *Proceedings of the 2006 Conference on New interfaces for musical expression (NIME '06), Paris, France*, 2006.

[5] N. Marwan, M. Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5-6):237–329, 2007.

[6] G. Roma and A. Xambò. A tabletop waveform editor for live performance. In *Proceedings of the 2008 Conference on New interfaces for musical expression (NIME '08), Genoa, Italy*, 2008.

[7] S. Streich and B. Ong. A music loop explorer system. In *Proceedings of the 2008 International Computer Music Conference (ICMC), Belfast, Northern Ireland*, 2008.

Designing the EP trio: Instrument identities, control and performance practice in an electronic chamber music ensemble

Erika Donald
Centre for Interdisciplinary Research in Music Media and Technology, McGill University
Montréal, Canada
erika.donald@mail.mcgill.ca

Ben Duinker
Centre for Interdisciplinary Research in Music Media and Technology, McGill University
Montréal, Canada
benjamin.duinker@mail.mcgill.ca

Eliot Britton
Centre for Interdisciplinary Research in Music Media and Technology, McGill University
Montréal, Canada
eliot.britton@mail.mcgill.ca

ABSTRACT

This paper outlines the formation of the Expanded Performance (EP) trio, a chamber ensemble comprised of electric cello with sensor bow, augmented digital percussion, and digital turntable with mixer. Decisions relating to physical set-ups and control capabilities, sonic identities, and mappings of each instrument, as well as their roles within the ensemble, are explored. The contributions of these factors to the design of a coherent, expressive ensemble and its emerging performance practice are considered. The trio proposes solutions to creation, rehearsal and performance issues in ensemble live electronics.

Keywords

Live electronics, digital performance, mapping, chamber music, ensemble, instrument identity

1. INTRODUCTION

Formed in late 2009, the EP trio is a small ensemble dedicated to research, creation, and performance in live electronic music. The trio is comprised of a unique combination of commercially available electronic instruments and equipment: electric cello with sensor-enabled bow controller and volume pedal, digital drum kit augmented with real-time DSP controller and amplified acoustic percussion, and digital turntable-based electronics. The group is focused on artistic applications of existing technologies within an ensemble framework.

In designing individual instrument identities and forging relationships between them, the EP trio draws on both Western classical chamber music (e.g., piano trio) and rock / pop / jazz "band" models. Sonically, the group blends contemporary electroacoustic and electronic music aesthetics. Its initial aim was to establish a streamlined set-up for small ensemble live electronic performance, emphasizing musical flexibility and technical self-sufficiency. By working within a partially fixed medium (i.e., fixed hardware and software), the trio explores this framework in depth, experimenting with various approaches to achieving compatible instrument control capabilities, sonic identities and gesture-sound mappings. This paper presents the challenges met by the EP trio in the creation of new repertoire and emergence of a performance practice for an electronic chamber music ensemble [2] [3] [8] [9].

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. ESTABLISHING SET-UP AND CONTROL CAPABILITIES

R. Murray Schafer coined the term *schizophonia*, meaning "split sound", to describe the disconnect between original acoustic sounds (i.e., those coupled to their physical production mechanisms) and their reproduction in other times or places [10]. The instruments of the EP trio possess highly contrasting sound (re)production and manipulation capabilities – they are *schizophonic* to varying degrees and in different ways. Thus, a challenge facing the group was to design instrumental set-ups that provide the expressive musical control and interaction possibilities necessary to function as a cohesive ensemble.

2.1 Individual Instrument Set-ups

Miranda and Wanderley define a digital musical instrument (DMI) as having three components: a sound source (synthesis), an interface (sensor input), and a mapping configuration relating these two [6]. The EP trio has carefully selected and combined newer and established commercially available equipment to enable diverse and complementary sound creation and control capabilities for each of the three performers. In the cello and percussion set-ups, the group sought to greatly expand the sonic palettes of acoustic instruments while utilizing many aspects of acoustic performance techniques. This section outlines the hardware and software employed by the EP trio and the reasoning behind these decisions.

2.1.1 Cello Set-up

The cello set-up is built around a Zeta Strados electric cello, chosen for its sound quality. Its active preamp is powered by a StringPort Polyphonic Stringed Instrument to USB2 Converter [4] that sends a polyphonic audio signal to a laptop. The cello is played with a K-Bow, a wireless sensor-bow that measures several bowing parameters and communicates via Bluetooth to the K-Apps software[1] [5]. Measured bowing parameters are: grip force, hair tension, 3D acceleration and tilt, and length and distance from IR and RF emitters, respectively, attached under the cello fingerboard. The K-Bow is a gestural controller and becomes a DMI when its sensor input is mapped to a sound source: it may be used to control processing and playback of live audio and/or samples in K-Apps. The K-Bow/K-Apps thereby adds an extra

[1] K-Bow and K-Apps Manual: www.keithmcmillen.com

layer of gesture-sound control capability to the cellist's performance. Every bowing gesture may perform two tasks, controlling not only sound production on the cello but also continuous processing parameters. This presents significant mapping and compositional challenges, which are addressed in 3.2.2 and 4.

The limited amplitude range of the electric cello was initially problematic in the ensemble. A Roland EV-5 Expression Pedal[2] was added as a volume controller to expand dynamic range and enable abrupt changes, allowing the cello to better match the ADSR (i.e., envelope) characteristics of the turntable and V-Drums. The volume pedal controls the main output of the K-Apps software, attenuating the audio signal sent from the cello set-up to the mixer.

2.1.2 Percussion Set-up

The percussion set-up is built around a 4-piece Roland V-Drum kit[3] with a TD-9 sound module used as a MIDI interface. This V-Drum kit was selected because it is portable, reliable, and provides tactile feedback similar to acoustic drums. Samples are triggered by velocity-sensitive drum pads and processed by a laptop running Native Instruments' Kontakt sampler[4]. Various DSP parameters are mapped to the sliders and knobs of a Korg NanoKontrol[5]. This set-up allows the percussionist to activate and modify samples in an intuitive and precise manner.

V-Drum cymbal pads are replaced with acoustic cymbals. This set-up is augmented with small, resonant acoustic percussion instruments (e.g. bowed crotales), amplified to suit the performance space and processed by the turntablist at the mixer (see Figure 1). These modifications provide acoustic sound options that expand the percussion sound palette and are often used to enhance blend with the cello sounds.

2.1.3 Turntable Set-up

The turntable set-up consists of a Pioneer CDJ-1000 MK3[6] digital turntable and an Allen and Heath Xone 92 mixer[7]. The CDJ-1000 MK3 was selected for its robustness, portability, and wide feature set. Its control features are modeled on those of standard vinyl turntables, including touch sensitive platters, master pitch/tempo controls and brake. The CDJ also has adjustable brake speeds, reverse, and expanded pitch range and cue options, which enable greater manipulation of audio materials.

Manipulating and mixing multiple sound sources in real time is the core of DJ performance practice. In the EP trio, the cello and percussion set-ups take the place of additional turntables that might be routed to the mixer and adjusted by the turntablist. Musical use of EQ, cross fading, mixing and effects processing, applied to the ensemble as a whole, help to achieve balance and blend, resulting in a cohesive ensemble sound.

2.2 Ensemble Set-up

Figure 1 depicts the hardware set-up and signal flow of the EP trio. The dotted line represents a Bluetooth connection.

2.2.1 Ensemble Sound Scheme

As shown in Figure 1, sound is mixed within the ensemble (by the turntablist) and a single mix output as a stereo signal to both house and onstage monitors. This decision was motivated by the desire for technical autonomy (i.e., no need for a sound person); the trio retains control of onstage monitor levels. Decisions not to use individual mixes, headphones, or spatialized monitors were prompted by performers' wishes to keep physical set-up simple and to hear the same mix as the audience, thereby developing control of their own sounds as part of the ensemble (much like an acoustic chamber group). These choices present major implications for the design of individual instrument sounds, identities and roles within the group, and their means of control – sounds must be reasonably distinct and performance gestures clear (as noted in Donald's previous ensemble DMI performance experience)[9].

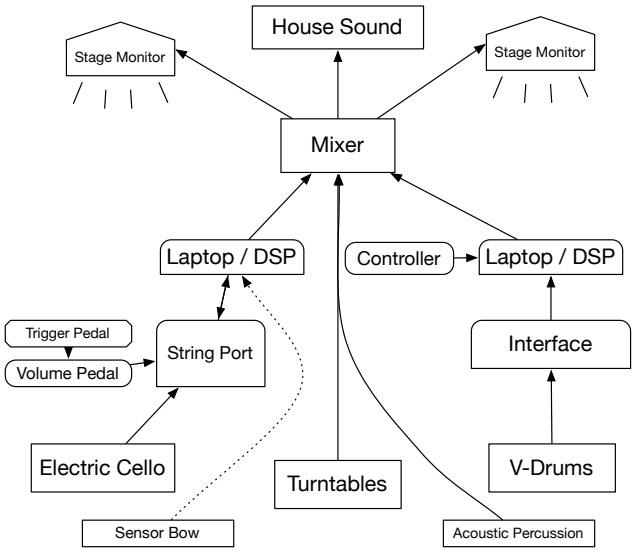


Figure 1: EP trio set-up and signal flow.

3. CREATING INSTRUMENT AND ENSEMBLE IDENTITIES

Individual instrument identities, and roles within the EP trio, are dependent on decisions relating to all three components of DMIs described by Miranda and Wanderley. These are: the physical interfaces and control capabilities they afford; the sounds produced, whether inherent (acoustic) or assigned (sampled, synthesized or processed); and the mappings between the two. The fixed set-up of the EP trio was described in the previous section. In the present section, the contributions of sounds and mappings to the development of instrument identities and roles, and finally the emergence of an ensemble performance practice, will be considered.

3.1 Instrument Identities and Mappings

In a context where each of the instruments can sound like (almost) anything, they begin to be defined by their control capabilities and limitations. Individual instrument identities will become more defined as a larger body of repertoire for the EP trio is created, revealing which sound and mapping elements are particular to specific compositions and which are consistently retained by each instrument.

The decision to hear only one mix from shared monitors results in additional challenges (described in 2.2.1). When instrument sounds are similar and their source shared, performers have difficulty distinguishing their own sounds, resulting in diminished control [9]. It is preferable that each instrument's sounds be distinct, however, the ability to blend

[2]Roland Corporation. www.roland.com/products/en/EV-5
[3]Roland Corporation. TD-9 V-Drums, Japan. (2008)
[4]Native Instruments. Kontakt. Berlin, Germany. (2010)
[5]Korg, Inc. NanoKontrol. Tokyo, Japan. (2008)
[6]www.pioneer.eu/uk/products/archive/CDJ-1000 MK3
[7]www.allen-heath.co.uk/uk/xone92.asp

may be essential to some compositions. In these instances, control/performance gestures must be very clear. Therefore, the EP trio is developing a core of gesture-processing mappings that remain quite consistent, despite changing sonic materials (see Figure 2). These include gestures to control volume, envelope shape and aspects of timbre for each instrument, with particular attention to filtering, amplitude control, velocity scaling and gain staging. This provides a means for performers to play expressively and "together" as an ensemble – dynamic and timbral ranges are controllable and compatible. A stable set of mappings also establishes mutual understanding of the control resulting from performance gestures, providing some degree of multimodal congruence between gesture and sound. Additional mappings may vary from one composition, or moment, to the next according to the need to control specific sonic materials (see Figure 2).

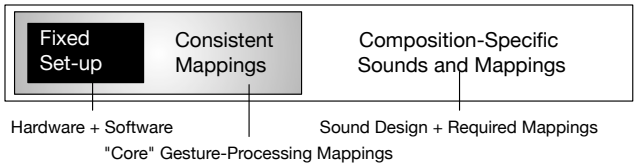


Figure 2: Core and composition-specific mappings.

3.1.1 Cello Identity

The electric cello is the least *schizophonic* in the ensemble – its sonic identity is largely tied to its acoustically generated sound. Like the electric guitar, it produces an amplified audio signal that may be modified by applying various digital signal processing (DSP) effects. In the EP trio, the cello is the only instrument able to accurately perform pitched material, suggesting a role as a melodic instrument. The K-Bow/K-Apps enables playback of samples, and continuous control of sound processing by bowing gestures, potentially allowing the cello to assume other roles.

3.1.2 Cello Mapping

The K-Bow/K-Apps adds a dual layer of gesture-sound control capability to the cellist's performance: bowing gestures may determine both sound production on the cello and continuous processing parameters. Achieving compatibility between the two presents significant mapping and compositional challenges. The cellist and/or composer specify the types of DSP and their control (and scaling) by assigning sensed bowing parameters to effects (e.g. bow length to delay; distance from fingerboard to filter frequency.) in K-Apps. Sound production and control gestures must be congruous, highly repeatable, reasonably intuitive, and ergonomic to the cellist. This necessitates close collaboration between cellist and composer and careful compositional planning. Infinite mappings are possible, but most successful combinations draw on established cello techniques.

3.1.3 Percussion Identity

The identities and roles of the percussion set-up in the EP trio are defined by the sample libraries controlled and the limitations imposed by trigger-based performance. Despite these constraints, it can produce both event-based and textural sonic materials, using sample manipulation parameters and acoustic percussion. To fulfill these roles, appropriate sample library construction is critical. Much of the percussion's sonic identity is re-created with each piece.

3.1.4 Percussion Mapping

Logical sample to drum pad assignment is paramount for the V-Drums. The standard drum kit organizes drum location based on pitch range: lower sounds are located at the player's right (floor toms) and feet (kick drum)[8]. As the kick drum is typically the lowest pitched instrument, the kick drum pad is reserved for lower-pitched or loop-based samples, utilizing that drum's association with keeping steady time. Pitch, envelope, and filters are modified in real time via NanoKontrol, mapped according DSP required.

3.1.5 Turntable Identity and Mapping

The limitations and idiosyncrasies of the turntable define its identity within the EP trio. It has fixed mappings of performer gesture to control/manipulation parameter but the sound materials available may change from moment to moment and piece to piece. The sounds the turntable (re)produces are samples created by the composer or selected by the performer. Textural drones, rhythmic materials, events, transitions, scratch solos and pre-recorded tape passages are musical/structural roles the turntable can readily perform.

3.2 Flexibility of Sounds and Mappings

Despite arriving at a fixed set-up, the wealth of sound and mapping possibilities for each instrument is only partially constrained. Some sounds are inherent to an instrument (i.e. acoustic percussion, electric cello) but may be modified through digital processing. Other sonic materials are entirely at the discretion of the performers and/or composers (i.e. those reproduced by sample-based instruments: turntables, V-Drums, and potentially K-Bow). The instruments of the EP trio also differ greatly in their capacity to alter mappings, thus ranging from minimal to total flexibility in both their sonic and mapping possibilities. Each instrument's sounds and the performance gestures/playing techniques enabled by its mapping(s) contribute to its individual identity and role(s) within the ensemble.

3.2.1 Two Dimensions of Flexibility

As represented in figure 3, the cello set-up has the least flexibility in its sonic identity (when not using the K-Bow as a sample playback controller) while the percussion has the most, as it can readily trigger more simultaneous samples than the turntable. However, the cello set-up affords highly flexible gesture-processing mapping strategies because a multitude of DSP parameters can be mapped to a number of continuous sensor input streams from the K-Bow (and combined with live creation of audio material). In contrast, the mapping of turntable control gestures to sound processing is essentially pre-established and fixed. The EP trio percussion set-up falls somewhere in the middle in its mapping flexibility – though its samples must always be triggered by a striking gesture, both samples and processing effects are flexibly assigned to the drum pads and the NanoKontrol's knobs and sliders, respectively. Because of its high flexibility in terms of both sonic identities and mappings, the percussion set-up often serves as the "glue", or mediator, between the cello and turntable set-ups, providing a middle ground between the two.

4. REHEARSAL AND CREATIVE ISSUES

Numerous, interrelated challenges encountered during the creation and rehearsal of new works continue to influence the development of the EP trio's instruments and the emergence of an ensemble performance practice.

[8]Based on normal kit set-up for right-handed drummer

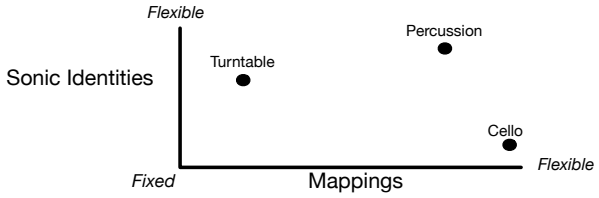


Figure 3: Flexibility of sonic identities and mappings.

4.1 Latency

With practice, all performers can learn to adjust for a certain amount of latency, but too much or variable/random latency can destroy performer and audience perception of sounds resulting from performance gestures (i.e. multisensory congruence). The V-Drums and turntable have imperceptible to minimal latency. However, the cello is affected by variable latency depending on the DSP applied. To reduce the negative effects of this issue [7], cello parts with demanding processing are composed primarily of sustained sounds and textural effects that do not require rhythmic precision. Latency challenges are resolved through collaboration between composer and performers on successful musical materials and careful adjustment of DSP and mappings, and by individual practice.

4.2 Ensemble Performance and Expressivity

Successful negotiation of latency issues allows the EP trio to overcome synchronization difficulties. Oore advocates technical mastery on new instruments [8] – this is of utmost importance in an ensemble setting. Not only must performers be highly proficient and consistent on their own instrument, they must be flexible enough to adapt in real time to nuances in colleagues' performance. It has been extremely helpful for members of the EP trio to understand the limitations and capabilities of each other's set-up so they may anticipate and react to best effect, resulting in "tight" ensemble performance and enhanced musical expressivity.

4.3 Making Changes in Rehearsal

Working with DMIs in an ensemble context presents a special rehearsal challenge: namely, that sample-based materials, DSP effects and mappings cannot be instantly modified. This limitation influences the sonic materials, compositional structures and mapping strategies used by composers and performers. Building flexibility into patches, sampled materials and compositions can allow for some on-the-spot tweaking. Minor adjustments become an important part of the rehearsal process, however major changes require work outside of ensemble rehearsal. For each performer, a thorough knowledge of the sound creation and mapping processes behind their instrument is invaluable as it allows for rapid troubleshooting and clear communication with composers and colleagues.

4.4 Composition Process

Composers writing for the EP trio must either accept previously established sound palettes and gesture-processing mappings and rework these into a new composition (as they would in writing for acoustic instruments), or create new sonic materials and mappings. In the latter case, it is preferable that composers work closely with ensemble members to develop comfortable and effective means of controlling sounds. However, despite collaboration, developing and learning to perform with new mappings require time and practice and will invariably increase the scope and duration of a compositional project.

5. CONCLUSIONS

The EP trio has achieved its goal of establishing a robust and musically expressive ensemble. Through various approaches to developing instrument identities, mappings, and performance practice, the group has integrated three contrasting instrumental set-ups. Careful attention to musical identities and limitations has laid the foundation for a viable and musically satisfying electronic chamber music ensemble

5.1 Future Directions

The EP trio will expand its repertoire and commission outside composers. This collaborative process will provide new perspectives though the practical and artistic challenges encountered in each project. During creation, rehearsal, and performance phases, the ensemble's "performance practice" will continue to evolve. The trio plans to assemble the findings of these processes into a detailed set of compositional and procedural instructions that may prove useful to others working with live electronics in chamber ensemble settings.

6. ACKNOWLEDGMENTS

The EP trio gratefully acknowledges the support of the Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT) and the Digital Composition Studios (DCS) at the Schulich School of Music of McGill University. Special thanks to: our supervisors Isabelle Cossette, Sean Ferguson, Fabrice Marandola, and Marcelo Wanderley; Richard McKenzie (DCS) for technical assistance; Parker Bert, the group's original percussionist; and CIRMMT staff.

7. REFERENCES

[1] Bongers, Bert. (2000). Physical Interfaces in the Electronic Arts: Interaction Theory and Interfacing Techniques for Real-time Performance. In M.M. Wanderley and M. Battier, eds. *Trends in Gestural Control of Music*. Ircam – Centre Pompidou: 41-70.

[2] Jorda, Sergi. New Interfaces and New Music-Making Paradigms. *NIME Proceedings*, Seattle, USA, 2001.

[3] Kimura, Mari. (2003). Creative process and performance practice of interactive computer music: a performer's tale. *Organised Sound*, 8(3): 289-296.

[4] McMillen, Keith A. Computer Input Device for Polyphonic Stringed Instruments. *ICMC Proceedings*, New York, 2010.

[5] McMillen, Keith A. Stage-Worthy Bows for Stringed Instruments. *NIME Proceedings*, Pittsburgh, PA, 2009.

[6] Miranda, E. R., and M. M. Wanderley. *New Digital Music Instaruments: Control and Interaction Beyond the Keyboard*. Middleton, WI: A-R Editions, 2006.

[7] Nagashima, Yoichi. Measurement of Latency in Interactive Multimedia Art. *NIME Proceedings, Hamamatsu*, Japan, 2004.

[8] Oore, Sageev. Learning Advanced Skills on New Instruments. *NIME Proceedings*, Vancouver, Canada, 2005.

[9] Pestova, X., E. Donald, et al. (2009). The CIRMMT/ McGill Digital Orchestra Project. *ICMC Proceedings*, (pp. 295-298), Montreal, Canada, 2009.

[10] Schafer, R. Murray. *The New Soundscape*. Don Mills, ON: BMI Canada, 1969.

[11] Tanaka, Atau. Mapping out Instruments, Affordances, and Mobiles. *NIME Proceedings*, Sydney, Australia, 2010.

Perceptions of Skill in Performances with Acoustic and Electronic Instruments

A. Cavan Fyans, Michael Gurevich
Sonic Arts Research Centre
Queen's University Belfast
BT7 1NN, UK
{afyans01, m.gurevich}@qub.ac.uk

ABSTRACT

We present observations from two separate studies of spectators' perceptions of musical performances, one involving two acoustic instruments, the other two electronic instruments. Both studies followed the same qualitative method, using structured interviews to ascertain and compare spectators' experiences. In this paper, we focus on outcomes pertaining to perceptions of the performers' skill, relating to concepts of embodiment and communities of practice.

Keywords

skill, embodiment, perception, effort, control, spectator

1. INTRODUCTION

The subjects of skill and virtuosity in Digital Musical Interactions (DMIs) [11] have emerged as prominent concerns in NIME. The literature reflects a desire for DMIs that can support virtuosity in performance [24, 19, 2], of which skill is an important component [12]. However, we have asserted that skill is not a quantity contained solely within the interaction between the performer and instrument, but exists also in a wider context that includes subjective assessments made by spectators [9, 13, 4, 15].

A previous study exploring spectators' understanding of performance with DMIs addressed mental models and the understanding or error [10]. Results from this study challenged the assumption (or failure to question) that audiences would perceive performances with electronic instruments in the same manner as those with traditional instruments [11]. Skill development with DMIs has largely been accepted to function in much the same manner as acoustic instruments [19, 2], in spite of the fact that others have described inherent differences in developing skills with digital technologies [6], attributable in part to the disembodied nature of interaction with many digital systems [8].

The broader HCI literature has begun to address questions of skill in digital interactions, but, with very few exceptions [20, 5, 14] it has not taken spectators into account. Discussions of the *perception* of skill are even rarer still. Djajadiningrat, Matthews and Stienstra [6] describe the aesthetic value of skilled action for both the actor and spectators, but do not delve into the specifics of how skill is perceived. Using point-light displays, Rodger [21] examined the role of bodily movement in spectators' ability to discern the skill level of clarinet performers, however this study focused primarily on acoustic music in known contexts.

In NIME, in spite of the widespread desire to see more virtuosic performances, there has been very little discussion of what actually constitutes skill in performances involving DMIs, nor of the spectator's contribution to this determination of skill. Studies in the literature tend to ask the question "*how can I become more skilled on this instrument?*" rather than "*why does a spectator think I'm skilled?*".

Following a previous study examining spectators' understanding of performance with DMIs [10], we conducted a qualitative study of acoustic instruments following the same methodology. It is important to note that we are not evaluating the instruments in the study, nor using them to draw generalized distinctions between perceptions of acoustic and electronic instruments. This paper presents observations from both studies, from which we identify phenomena that underlie the perception and understanding of skill.

2. METHODOLOGY

Twenty seven participants were selected to take part in the study. Each participant was individually shown two short video performances. One was an original contemporary composition for solo violin (*Broken Flames and Little Wind* by R. Mannion), performed by a professional violinist with approximately fifteen years experience. The piece explored timbral and textural variation through combinations of standard and extended technique. The second performance was a solo structured improvisation with the *sheng*, a Chinese blown free-reed instrument. This was performed by a PhD student in computer music with nearly twenty years of practice on the saxophone in jazz and free improvisation. The performer had only had a few hours of experience with the *sheng* before the performance was recorded.

After viewing the performances participants were prompted to discuss a variety of aspects of the performance in a structured interview. In this paper we focus on their discussions of skill. The interviews were recorded on video for post-study transcription and analysis. The method of presentation and analysis was the same as in the previous study – full details can be found in [10].

The violin and *sheng* were selected in order to reflect aspects of the instruments (Theremin and Tilt-Synth) used in the previous study [10]. A counterpart to the Theremin, the violin was selected in order to ensure that all participants would be familiar with the instrument and broadly how the performer's actions correlated to the resultant sounds. As with the The Tilt-Synth, which was created specifically for the prior study, the *sheng* was chosen primarily as an instrument that very few people would be familiar with. The results confirmed that only one participant had prior experience with a *sheng*. Although we selected instruments that

NIME'11, 30 May–1 June 2011, Oslo, Norway.

we suspected would give rise to diverse spectator experiences – we expected that familiarity would have an impact on perceptions of skill – we emphasize that this was a qualitative study; the instruments, performances and performers' skill were not treated as controlled independent variables as one would in a quantitative experiment. Rather, we observed a range of participants and performers, and through rigorous qualitative analysis of participant interviews, identified common phenomena that underlie spectators' perceptions and experiences of skilled performance.

3. OBSERVATIONS

We present observations and analysis from both studies that relate specifically to participants' perceptions and assessments of skill. Two important themes emerged: those of embodiment and communities of practice.

3.1 Skill as Embodied Interaction

The concept of embodied interaction has gained prominence in HCI and cognitive science, more recently applied to musical instruments [1, 3]. Described as an intrinsic coupling between an agent and its environment [16], embodied interaction denotes "participative status" in the unfolding of an activity [7]. Embodied interaction thus depends on immersive, situated and timely engagement [1, 17], in which action is inextricably linked to perception that is guided by biological, psychological and cultural forces [23]. According to Dourish, "embodiment is about engaged action rather than disembodied cognition; it is about the particular rather than the theory, directness rather than disconnectedness" [8].

A central premise in much of Ingold's work is that skill is embodied knowledge [15]. In our observations, it is clear that participants perceived it as such; the perception of (dis)embodiment in the performer's interaction with their instrument featured prominently in descriptions of skill across all performances. Furthermore, participants' embodied knowledge from their own hands-on experiences with musical instruments was central in forming perceptions of skill among the performances they saw.

3.1.1 Confidence

In describing the performers' skill in both of our studies, a perception of physical confidence or comfort was salient for many participants. Many such comments (violin: n=10, *sheng*: n=5, Theremin: n=5, Tilt-Synth: n=4) were generic, such as, "*She was very confident with the instrument*," or "*She looked very comfortable with what she was doing.*" However, a number of participants went further; one noted that the violinist "*looked quite natural... It wasn't like she was sitting there thinking about her technique in any way.*" Another participant made a similar observation of the *sheng* performer: "*He had his eyes closed and he wasn't trying to watch where his fingers were going, so he must have known the instrument rather intuitively.*" The participants describe a state in which the performer is not actively attending to their instrument; they are not exploring or playing *with* the instrument, but rather creating sounds *through* it.

Thus, more than just 'confidence', participants appeared to sense an embodied connection between the performer and their instrument. This was frequently articulated as the initial, and in some cases the only, influence on judgements of skill. It is important to highlight that this impression was not solely based on posture of physical comportment; several participants described perceiving confidence in sound and action. Many perceived the performer's embodied connection to their instrument holistically. Whereas we suspected that participants might piece together evidence of skill from individual perceptible features (errors, slips, technical facility), it appears that skill is an embodied impression, perceived 'directly', as ecological psychology would suggest [4]. Note the way the following participant interchangeably describes visual and sonic features in their assessment of the *sheng* player's skill:

> "*So there were some notes that I felt like 'I'm not sure if he meant to play that.' I feel like he was still exploring the instrument and its potential... I think in some ways it also looked like he may not have known - it sounded like it was more of an experiment. Like, 'Oh god I hope this comes out the way I want it to.'*"

3.1.2 Disembodiment

Positive descriptions of skill in terms of confidence were more frequently associated with the violinist, who was the only performer among the four who had substantial experience on the instrument they played for the study. In contrast, assessments of the *sheng* performer's skill in these terms were ambivalent. Although some took note of his confidence, others described a perception that the performer was concentrating on finding his way around the instrument. Thus, failing to engage with it in an embodied way was indicative of a *lack* of skill. In the DMI study, although a number of participants also described those performers' skill in terms of confidence, a sense of disembodiment was also associated with negative impressions of skill. Of the Thereminist, one participant said, "*He was concentrating on it, he had his eyes on both his hands and the antennae as well.*" Another judged the Tilt-Synth performer to be unskilled because "*he looked very self conscious at the start... He wasn't confident.*"

3.1.3 Perception as Embodied Experience

When discussing skill and difficulty in both studies, many participants' descriptions were based on personal experiences with musical instruments, suggesting they experienced the performance in terms of their own embodied knowledge. For the violin, many (n=8) focused on experience with the violin itself. Praising the violinist's skill, one participant recalled, "*I've held a violin and bow in my hand, it was too small and the bow was too awkward!*"

Participants also referred to a *lack* of personal experience with instrument in the study as contributing to an inability to assess skill. Five participants highlighted that they don't play the violin. For the *sheng*, two participants said they could not judge the performance because they had never played or held the instrument. According to one: "*I'd have to play the [sheng] myself to see. I couldn't gauge. Whereas I know how a violin works, so I thought her performance was very skilful.*" Regarding the Tilt-Synth, one participant said they would "*have to play it in order to make judgements.*"

However, many who lacked direct experience with the instruments in the performances expressed their perceptions of skill in terms of other instruments they *had* played. Of the *sheng*, one noted, "*I've tried to play a clarinet before and I know it takes a bit of skill, it's not just blowing, you have to blow a certain way.*" One participant related the violin's difficulty to his experience as a guitarist: "*Well obviously, compared to a guitar, it doesn't have frets so you don't know where to put your fingers.*" Of course, the violinist *does* know where to put *her* fingers – she plays the violin! The guitarist is assessing the performance as if he were playing the violin, in which case, as a guitarist, he wouldn't know where to put his fingers. This tendency was far more prevalent with the acoustic instruments than with the electronic ones. Notably, no participants described the Tilt-Synth in terms of their own instrumental experience.

3.1.4 Control and Effort

Perceptions of control and effort frequently appeared in participants' discussions of skill. For the violin, comments (n=14) focused on the accuracy of manual control necessary to produce specific pitches. Many (n=12) similarly described the difficulty of the Theremin in terms of control and precision of hand/arm movement. Several participants even related control of the Theremin to the violin (recall these were separate studies). One described the Theremin as "*kind of like the violin but more difficult*," due to the precision required to achieve specific pitches. Unlike the violin, participants (n=8) discussed the difficulty of the Theremin as a matter of coordination between both hands. One participant described it as "*rubbing your tummy and patting your head... I imagine its hard to do both things.*"

In contrast to all the other instruments, participants did not describe specifically physical challenges to playing the Tilt-Synth. Many (n=9) echoed that the instrument was "*more complicated than the Theremin because there are more controls on it.*" However, this difficulty was seen as a cognitive or intellectual challenge, rather than a physical one. One participant asserted that it was "*the sheer amount of buttons he had to remember*" making the Tilt-Synth difficult. Furthermore, several participants believed that skill in playing the Tilt-Synth was a matter of technical knowledge of the system, rather than physically interacting with it. One participant commented that, "*In order to understand what's going on inside and make it sound the way you want it to, there's a fair amount of skill involved in that ... more technical, intellectual skill than physical performance skill.*"

In addition to control, *effort* was especially salient in assessments of skill in the *sheng* performance. One participant exemplified both perceptions stating, "*you would really need to be good at controlling your breath. You would have to have a very strong diaphragm.*" In contrast to the *sheng*, we observed that participants' comments regarding the Tilt-Synth alluded to a distinct lack of effort. Several specified that the Tilt-Synth was, "*simple to control*," that the performer was "*just pressing buttons.*"

3.2 Skill Exists in a Community of Practice

Lave and Wenger [18] describe the importance of what they termed a "community of practice" in activities where skill development is important. They claim that *identity* and *meaning* are imprinted into practical actions through the presence of a community of practice. Dourish similarly asserts that "in becoming a member of the community, one learns not only to exercise the skills of that community, but also to exercise them as a member of that community - with the same set of understandings, expectations, significances and meanings that are characteristic of that community and how it sees itself" [8]. From the perspective of a spectator, in order for the practitioner's skilled action to bear meaning the spectator must have knowledge of the community of practice in which it is situated.

3.2.1 Effect on Skill Assessments

A "lack of familiarity" dominated participants' comments in assessing the skill of the *sheng* and Tilt-Synth performers. For the *sheng*, these comments (n=23) are exemplified by statements like, "*It's hard to judge because I've never seen anyone else play that instrument.*" Another reported, "*There's no expectation of what it should sound like or what is proper Tilt-Synth playing.*" These participants did not simply highlight that they were unfamiliar with the instrument, but that they lacked a frame of reference in which to judge skill. They had no experience of a community of practice or prior exemplars that would imbue meaning to the performers' actions. This absence of a relevant community of practice resulted in divergent judgements or an inability to assess skill. Summarizing the difference between the violin and *sheng*, one participant stated, "*With the violin you are drawing from all the teachers and the wealth of knowledge about it...* [The sheng] *is hard to judge because I've never even seen anyone else play that instrument.*"

When participants were able to relate the performances to a body of experiences, they were more likely to formulate an assessment of skill based in part on the community from which these experiences were drawn. Several participants (violin: n=9, Theremin: n=8) discussed skill in reference to what they characterized as beginners or experts. When asked to describe the skill of the Thereminist, one participant replied, "*On a scale of 1 to Rockmore?*" Another focused on the performer's 'unconventional' technique, claiming "*95 percent of people don't play a Theremin like that.*"

Several participants further said they expected the Thereminist to deliver a violin-like performance. The expectation of a particular style of performance stemmed from having experienced highly skilled performances (e.g., Rockmore), but also from the timbre of the instrument. Frequent comparisons to the violin suggest that the distinctly string-like timbre of the Theremin gave rise to corresponding expectations of the performance.

In discussions of the *sheng*, several participants (n=10) associated skill with perceived errors. However, due to unfamiliarity with both the instrument and its associated performance practice – the absence of what it 'should' sound like – many could not conclude whether the perceived sonic artifacts were errors, an inherent and unavoidable part of the instrument, or intentional. One participant was confused whether perceived errors were a "*limitation of his ability or a limitation of the instrument. Or whether it was a conscious decision to have those bits that sounded like flaws.*"

Some participants unfamiliar with contemporary violin performance were surprised by the timbres the violinist employed (including harmonics, scratching and *col legno*). In many cases this led to difficulty judging skill, along with confusion or ambiguity between the expected sounds of a beginner and those of an expert. One participant, unsure of the performer's skill, deemed that the performance could have been a product of "*someone mucking about on the violin.*" Others placed this contradiction in a wider social context [18], associating the "*scratchy*" timbres with those of beginners or school concerts. Especially for the violin, participants appeared keenly aware of the entire continuum of skill in the overall body of practice, along which they were able to place this performer's skill almost instinctively.

4. DISCUSSION

We have observed that spectators perceive skill in musical performance as an embodied phenomenon. This gave rise to vastly different assessments of skill according to the diversity of experiences with the instruments employed in the studies. The violinist was by far the most experienced performer with the instrument she played in the study, also the most familiar among participants. Consequently, participants developed a strong impression of confidence or 'naturalness' in her interaction, even before she started playing. There was a perception that she *knew* the violin, not in an intellectual or technical way, but a bodily way. This is borne out by participants with a high estimation of her skill but difficulty expressing why: "*She appeared to be certainly classically trained; just the precision and rigidity and that kind of thing. Her touch on the violin – she had obviously been practising and knew what she was doing on the violin.*"

The *sheng* player was an expert saxophonist and improviser, but was new to this instrument. Many participants perceived musical knowledge manifested in his performance, but did not see the same bodily facility as with the violinist. With the Tilt-Synth, there was a similar impression of *disembodiment*; that the performer was 'exploring the instrument.' We do not claim that these impressions of the Tilt-Synth are necessarily characteristic of all DMIs; the performers in our studies had varying degrees of experience with their instruments, which were not intended to represent all acoustic or electronic instruments. Rather, we highlight the centrality of embodiment – the perceived *disconnect* between performer and instrument was as salient for spectators as the violinist's embodied engagement – and note that many authors have identified disembodiment as a particular and significant challenge in digital interactions [6, 8, 17, 22].

Many participants' descriptions of skill reflect Heidegger's distinction between *ready-to-hand* and *present-at-hand*, seen as a foundation for the theory of embodied interaction [8]. *Ready-to-hand* describes a state of interaction that is 'action without theorizing' in which an object becomes an invisible extension of the user. Klemmer [17] identifies a human capability characterized by "the intimate incorporation of an artifact into bodily practice to the point where people perceive that artifact as an extension of themselves; they act *through* it rather than *on* it." Significantly, this capability was perceptible to spectators in our studies, and was among the most salient phenomena in their discussions of skill.

Our studies also revealed that spectators perceived skill in terms of their own bodies; experiences with musical instruments they had played were particularly important. Participants' embodied knowledge, or lack thereof, led to significant differences in perceptions of skill for the violin and *sheng*. Whereas some participants' personal experiences of the difficulty of playing the violin (or other stringed instruments) led to high estimations of the performer's skill, the lack of embodied knowledge of the *sheng* (or anything similar) confounded their ability to assess it.

We observed corresponding differences between the Theremin and the Tilt-Synth. Part of the difference is attributable to the relative novelty of the Tilt-Synth; whereas some participants were able to situate the Theremin performance within a body of known practice or in terms of their own experience, this was impossible for the Tilt-Synth, which no participant had ever seen.

Elsewhere we proposed that there is something deeper characterizing the differences between the Theremin and the Tilt-Synth [11]. Even among those with little prior exposure to the Theremin, there was a stronger tendency to understand the performance in terms of other musical instruments or skilled actions. There was a greater sense of *instrumentality* with the Theremin; it was more strongly associated with the violin than with the Tilt-Synth. This is further brought to bear by participants who dismissed the Tilt-Synth performance as mere "button-pressing." Participants ascribed physical difficulty, exertion and necessity for control to the Theremin, whether the performer was able to achieve it or not. In contrast, perceptions of the Tilt-Synth reflected a lack of effort; participants perceived rich and diverse sounds, yet a simple physical interaction, and thus attributed skill to the performer's intellectual understanding of the instrument rather than embodied knowledge.

5. REFERENCES

[1] N. Armstrong. *An Enactive Approach to Digital Musical Instrument Design*. PhD dissertation, Princeton University, 2006.

[2] J. Butler. Creating pedagogical etudes for interactive instruments. *Proc. NIME*, 2008.

[3] R. Chaffin and T. Logan. Practicing perfection: How concert soloists prepare for performance. *Advances in Cognitive Psychology*, 2(2):113–130, Jan. 2006.

[4] E. F. Clarke. *Ways of Listening: An Ecological Approach to the Perception of Musical Meaning*. Oxford University Press, 2005.

[5] P. Dalsgaard and L. K. Hansen. Performing perception: staging aesthetics of interaction. *ACM TOCHI*, 15(3):13:1–13:33, 2008.

[6] T. Djajadiningrat, B. Matthews, and M. Stienstra. Easy doesn't do it: skill and expression in tangible aesthetics. *Personal and Ubiquitous Computing*, 11(8):657–676, 2007.

[7] P. Dourish. Embodied interaction: Exploring the foundations of a new approach to HCI. *CHI*, 2000.

[8] P. Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. MIT Press, 2001.

[9] A. C. Fyans, M. Gurevich, and P. Stapleton. Where did it all go wrong? a model of error from the spectators perspective. In *Proc. NIME*, 2009.

[10] A. C. Fyans, M. Gurevich, and P. Stapleton. Examining the spectator experience. *Proc. NIME*, 2010.

[11] M. Gurevich and A. C. Fyans. Digital musical interactions: Performer-System relationships and their perception by spectators. *Organised Sound*, 16(2), 2011.

[12] M. Gurevich, P. Stapleton, and A. Marquez-Borbon. Style and constraint in electronic musical instruments. *Proc. NIME*, 2010.

[13] M. Gurevich and J. Trevino. Expression and its discontents: toward an ecology of musical creation. In *Proc. NIME*, pages 106–111, 2007.

[14] C. Heath, P. Luff, D. V. Lehn, J. Hindmarsh, and J. Cleverly. Crafting participation: designing ecologies, configuring experience. *Visual Communication*, 1(1):9–33, 2002.

[15] T. Ingold. *The perception of the environment: essays on livelihood, dwelling and skill*. Routledge, 2000.

[16] J. H. Kim and U. Seifert. Embodiment and agency: Towards an aesthetics of interactive performativity. In *Proc. SMC*, pages 230–237, 2007.

[17] S. R. Klemmer, B. Hartmann, and L. Takayama. How bodies matter: five themes for interaction design. In *Proc. DIS*, pages 140–149, 2006.

[18] J. Lave and E. Wenger. *Situated learning: legitimate peripheral participation*. Cambridge, 1996.

[19] S. Oore. Learning advanced skills on new instruments. *Proc. NIME*, 2005.

[20] S. Reeves, S. Benford, C. O'Malley, and M. Fraser. Designing the spectator experience. In *Proc. SIGCHI*, pages 741–750, 2005.

[21] M. Rodger. *Musicians' Body Movements in Musical Skill Acquisition*. PhD thesis, Queen's University Belfast, 2010.

[22] D. Toop. *Haunted Weather*. Serpents Tail, London, 2004.

[23] F. J. Varela, E. Thompson, and E. Rosch. *The embodied mind : cognitive science and human experience*. MIT Press, Cambridge, MA, 1991.

[24] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

Cognitive Issues in Computer Music Programming

Hiroki NISHINO
Graduate School for Integrative
Sciences and Engineering
National University of Singapore
g0901876@nus.edu.sg

ABSTRACT
Programming Languages are the oldest 'new interface for music expression' in computer music history. Both composers and researchers in computer music still have considerable interests in computer music programming environments. However, while many researchers focus on such issues as efficiency, new paradigm, or new features in computer music programming, cognitive aspects of computer music programming has been rarely discussed. Such 'cognitive issues' are of importance when design or usability in computer music programming must be considered. By contextualizing computer music programming in the psychology of programming, it is made possible to borrow the technical terms and theoretical framework from the previous research in the field, which would be helpful to clarify the problems related to cognitive ergonomics and also beneficial to design a new programming environment with better usability in computer music.

Keywords
Computer music, programming language, the psychology of programming, usability

1. INTRODUCTION
Computer Music languages have been playing significant roles in musical creation since the birth of computer music in its history. Computer music programming is also very interesting in that computer music is at least one of the first fields, where a programming language was designed for artists as end-users, even when people hardly had access to computers. Even the design of *Music V*, one of the earliest computer music languages developed at Bell Telephone Laboratories, was enough `comprehensible` for musicians of that time without professional skills in computing, as seen in [14]. Since then, programming languages for musicians has been one of the main interests both from researchers and artists to explore the possibility of new territories in computer music.

Yet, the cognitive aspects of computer music programming have rarely been discussed in computer music community. The usability issues are seldom justified by the previous research in the psychology of programming and mostly supported only by the programming concepts or rather practical experience.

Such a lack in contextualization of the cognitive aspects of computer music programming can be significant obstacles for further research in usability issues.

By borrowing the technical terms and the theories from the previous research in the psychology of programming, the problems in computer music programming can be clarified so that the future research can be more beneficial to improve the designs of programming languages and environments for better usability in computer music programming activity.

2. RELATED WORK
In this section, we briefly describe the previous research in the psychology of programming so to contextualize computer music programming by the related work in the later section.

2.1 What is a Computer Program?

2.1.1 The surface structure and the deep structure
From a psychological point of view, *the surface structure* and *the deep structure* of a computer program must be distinguished. While the surface structure is about *textual structure* or how *surface units* are arranged in a program, the deep structure is based on the relations and the abstraction in a program, such as control flow, data flow and hierarchical organization of goal and sub-goals. A computer program is multi-dimensional in that it contains different types of deep structures.

2.1.2 Mental model
Mental model is a traditional approach in HCI to explain the understanding and reasoning by users about the system. Halsz and Moran's paper on mental models of a simple calculator is one of the traditional examples [12]. Mental model approach is also extended to programming languages. Détienne describes *"learning a programming language consists, therefore, in acquiring not only the syntax of language but the rules of operation of the virtual machine underlying it"* [9, p.17].

2.2 Program Design

2.2.1 Problem domain and computing domain
Program design has been studied mostly as problem-solving activity and considered to be composed of three phases; a programmer has to understanding a problem first. Then, research and development of the solution is conducted. Finally, he codes the solution. However, in the real world situation of programming design, programmers go back and forth between these phases as well as other design activities do.

2.2.2 Ill-defined problem
Program design activity is generally considered 'ill-defined', the characteristic of which is "*one that addresses complex issues and thus cannot easily be described in a concise, complete manner*" [18]. The goal of an ill-defined problem is often vague and some constraints and criteria may not be recognized at the beginning. For instance, a programmer may be able to notice that some specification is missing only after he started the design and in the course of implementing the solution for the specification, new constraints may be added to other parts of the problem.

Furthermore, there can be several different solutions for one ill-defined problem and there is hardly an objective true-or-false evaluation. Instead, the solutions can be evaluated by good/bad or appropriate/inappropriate assessments.

2.2.3 Software design activity

We briefly describe three different theoretical approaches to explain software design activity, i.e. knowledge-centered approach, strategy-centered approach and organization-centered approach. Detailed explanations can be found in [9, 13].

Knowledge-centered approaches focus on hierarchically organized knowledge stored in memory and programming activity is considered as activation of schemas; programmers utilize available schemas and combining them to solve the programming problems.

Strategy-centered approach focuses on the strategies that programmers take to solve the problem. For instance, in a problem that consists of hierarchically ordered sub-problems and sub-sub-problems, a programmer may work on top-down or bottom-up. The programmer may work on the end part of program first, then goes back to the beginning (forward vs backward development), or work bread-first or depth-first in hierarchical organization of sub-problems.

Organization-centered approach corresponds to the organization of the design activity and there are two models for this approach. One is the hierarchical model, influenced by structured programming, which models programming activity as problem solving of top-down, breadth-first searching process for a solution. On the other hand, the opportunistic model is based on the empirical studies on how a programmer deviates from hierarchical model; a programmer may write the part of the solution that they think is most crucial, not in top-down, breadth-first order. Green and his colleagues describe and explain such a behavior in [10].

2.3 Program Comprehension

2.3.1 Program Text Comprehension

As in programming design activity, several different approaches exist to explain program text comprehension. The theoretical framework of program text comprehension is largely based on natural text comprehension and there are several different approaches as in the case of programming activity.

In structural approach, superstructures (or a generic structure of a program) can play a significant role in comprehension process. Rist explains that the basic structure is made of input, calculate and output [19] and structural schema on such a basic structure guides the comprehension.

Détienne tried experimental validation of a functional approach, according to which program comprehension is processed top-down by activating knowledge schemas [8]. She also described the importance of mental model approach, in which "*to understand a program means to construct a detailed model of the situation"* as *"a theoretical approach that potentially has the predictive and explanatory power to account for how the comprehension activity is determined by the task"* and unlike the other models, mental model "*reflects the entities of the problem domain and their relationships, that is to say, the problem goals and the flow of data."*[9, pp93- 103].

2.3.2 Rules of discourse

Rules of discourse also play a significant role in program comprehension. Some rules of discourse can activate program schemas as in functional model in the previous chapter. Mullen tries to explain the importance of the rules of discourse by several other factors, such as chunks, split-attention effect, analogical reasoning etc. [15]. We pick up and briefly describe some of the examples by Mullen here below.

'Chunk' is "*a collection of memory elements having strong associations with one another, but weak association with elements within other chunks*" [15]. One of the rules of discourse that programmers share is separating each meaning full groups of code each other. Grouping the parts of the program together according to how mind chunks the related elements can help understanding of the code; e.g. the code can be easily understood if blank lines separate a group of four lines, which initialize one object, from the other part of the code.

Another rule of discourse is to keep the size of functions reasonable and not to distribute them sparsely in the different files as possible. Mullen explains this by the split attention effect, which makes the information difficult to comprehend by occurrence of indirection. For an example, *if text that supports a picture is presented separately from the picture it is more difficult to comprehend/learn than if the text were displayed meaningfully upon the picture itself* [15]. In a program text, if a part of code contains a lot of function calls to very small functions that are distributed among many different locations in the code, such a part of the code can cause lots of indirection and penalty for short-term memory, resulting in the split attention effect to decrease comprehensibility of the program.

Thus, the rules of discourse that programmers share can be also endorsed by the theoretical framework and play significant role in program comprehension.

2.3.3 Cognitive Fit

Cognitive fit theory developed by Vessey is the theory on the correspondence between the task performance and the representation format. For instance, *graphical representations emphasize spatial information while tables emphasize symbolic information* [21] and then a symbolic task can be performed better with tabular representation than with graphical representation and vice verse. Thus, fit and gap between a task and the representation of information is a significant factor in comprehension.

Some study reports the effect alike also in a textual programming language. Green showed *nested conditionals favored sequence information ("Given this input, what happens?)* and Gilmore and Green found that *a more declarative programming language gave improved access to circumstantial information* ("*Given this result, what do we know about the input?"*) [11].

2.3.4 Dual-task interference

Simultaneously working on two tasks can cause the interference between the given two tasks and the performance can be relatively worse than when each task is processed one after the other, not simultaneously [17]. Such dual-task interference has been observed between many different activities.

Shaft and Vessey considered the modification task of a program as dual-task interference situation and cognitive fit between comprehension and modification [20].

2.4 End User Programming

End-user software engineering or end-user programming is even considered as '*the most common form of programming in use today*' [2] and becoming an important research topic both in HCI and software engineering community. End-users who program for everyday work may not be expert in programming but they certainly are expert in their professions. Such an end-user is called a 'domain-expert end-user' or simply 'expert end-user'.

As Blackwell describes, "*an important characteristic of end-user programming research is that end-user programmers should not be regarded as "deficient" computer programmers, but recognized as experts in their own right and in their own domain of work. They might only write programs occasionally or casually, but it is possible that they have done so for many years'*, and thus the research on first year computer science students or the research on 'natural' programming languages by studying kids before learning any other language '*may not be directly relevant to needs of expert end-user programmers'* [1].

3. COGNITIVE ISSUES IN COMPUTER MUSIC PROGRAMMING ACTIVITY

In this section, we contextualize several aspects of computer music programming in the framework of the related work described in the previous chapters and also propose several interesting characteristics of computer music programming.

3.1 Program Design

3.1.1 Ill-defined problem, exploratory design, and the aesthetics of failure in computer music

Computer music programming also shares lots of characteristics with general programming activity and many problems in computer music are also ill-defined as in other programming activity. Yet the fact that the goal of a program design tasks is often composing a new computer music piece also bring some more interesting issues to be considered.

The constraints in ill-defined problems may be vague or even unrecognized at all when the program design activity is begun. Moreover, a goal of computer music programming is mostly a new computer music piece and this program design activity is highly exploratory a lot more than general programming tasks. A Composer may completely change the goal of the programming tasks; He might begin programming tasks with a short piece for tape in mind, but during his exploration, he may completely change the original plan and start writing for piano and interactive system. Even a bug or an error that a composer encounters can change the whole goal of the programming task. Cascone describes such a creative 'failure' in [4].

Such a highly exploratory design activity in computer music programming should be considered as a significant characteristic in designing a new programming environment.

3.1.2 Two languages in one environment

As mentioned in the previous section, a programmer is assumed to have the mental models of a device. One of the special characteristics in computer music programming, especially of textual computer music programming languages, is that they often mix two different programming paradigms into one language, each of which is based on a different mental model; while the synthesis models are normally declaratively defined, the other part of computer music programming language are usually based on different paradigm, such as instrument-score style, imperative programming or object-oriented programming.

While this feature may facilitate problem-solving on most of problems in computer music programming, it also may cause difficulty if the problems lies across the boundary of both domains of two languages.

3.2 Program Comprehension

3.2.1 Program Text Comprehension

Computer program is multi-dimensional and this is also true to computer music. Interestingly, computer music programming adds one more deep structure that is not in general purpose programming – musical structure, such as phrases, structures, forms, timbre, and the like. How to deal with this musical dimension should be highlighted as a significant factor in usability of computer music programming.

For instance, chunking the group of notes in one phrase in a c-sound score file may help the comprehension of the phrases so to recover the mental representation of the score, but such chunking also significantly damage to represent the relationship between the notes in different phrases; e.g. chunking one phrase in two voices of counterpoint makes it harder to grasp vertical relationship between the notes in two melodies while the melody in one voice can be clear described.

Recovering such deep structure of music contents in a program may be a difficult task, yet improvement in programming language syntax may be potentially beneficial to help recovering the musical representation from a program text.

However, if the musical contents are generated by certain algorithms and not explicit in the program text, it can even be almost impossible to imagine the musical output of the program, since mental or situational models related to musical events can be hardly recovered only by program texts.

3.2.2 Cognitive fit and cognitive styles

Carter and his colleagues described a *cognitive style* of composers in [3], relating it to the information processing strategy that the composers take. For instance, as for one of the characteristics called global/analytic, which corresponds to the composers' composition approaches; Those composers characterized as global tend to compose plan for the pieces they are working on, whereas other type of composers characterized as intuitive, in a more improvisatory approach. Such tendencies of global/analytic cognitive styles seem to correspond to the strategy-centered approach in design activity, such as top-down/bottom-up, breadth-first /depth-first strategy.

Dannenberg refers to cognitive styles in [7], to describe his work on the Nyquist composition environment, however, some aspects of the work seem to be more suitable in the framework of cognitive fit theory, rather than cognitive style. For instance, generally speaking, the shape of an ADSR envelope is much easier to grasp when it is visualized by a graphical representation than when it is described by a textual representation such as the list of floating-point values, whereas the exact duration of the the sustain in the same envelope is more comprehensive when the actual floating-point value is explicitly shown in the list, rather than estimating the duration by looking at the graphical representation of the envelope. Such cognitive ergonomics in graphical/textual representation can be easily explained by cognitive fit theory rather than by cognitive style.

Also as Green and his colleagues described in [10], programing activities by programmers in the real-world situations can be highly opportunistic. Such opportunistic behavior can be more significant especially when computer music programming is highly exploratory as described in the previous section. Even when a composer with global cognitive style work on the certain programming tasks, his programming activity can hardly be truly top-down.

Such issues as cognitive fit and strategic approach in programming activity should be considered important for further discussions on usability analysis of computer music programming environments.

3.2.3 Dual-task interference in live-coding

Live-coding would be an extreme type of computer music programming activity. Live-coding musicians perform their music, programming on-the-fly on the stage, sometimes even writing the code from the scratch. Nilson describes "*live-coding*

can demand producing functioning code to a strict time limit, to find ways to introduce or modify code with low latency" [16] and other paper by Collins and his colleague describe "*You forget the current audio or just take too long while you prepare the next section"* [6]. While the former description corresponds to the restriction on available time for coding given to a live-coding performer, the later also corresponds to the cognitive overload.

Such a nature of live-coding would be an unusual, but interesting case of dual-task interference. Certainly, listening to music in the professional level and writing code with the strict limitation in time are quite different mental activities, both of which consume considerable cognitive resources. Furthermore, modification task of existing code is often involved in live-coding performance and such modification task alone can be also considered as dual-task [20], as described in the previous chapter; The interference between multiple tasks can occur in live-coding and it is an interesting example to be discussed.

3.3 End User Programming

When placed in the thread of end-user programming, computer music programming is one of the most major, historical domains of expert end-user programming. Computer musicians are clearly an example of expert end-users, in that they have strong expertise in music domain but much less in computing. As described in [5], even in those days when the non-experts, who are not computer scientists, hardly had the access to computers, programming languages for computer music was being developed and composers with less expertise in programming had been invited to compose his musical pieces, using those tools and languages for computer music compositions. Furthermore, computer music programming is also an exceptional field even as expert end-user programming in that it already has a considerably long history and there are many end-users with the domain-expertise in music, a lot of who are educated in academic education of their expertise or with professional experience for many years.

Computer music programming as expert end-user programming activity also seems to be an ideal situation when we consider one of the traditional criticisms made to some of psychological studies on programming activity that the problem size is too small and far from the real world situations in which the programmers work in software industry. The problem in computer music is usually fairly small but still deals with the practical problems in their expertise domain of music.

4. CONCLUSION

Cognitive aspects of computer music programming have been rarely discussed in computer music community. Yet, by borrowing the theoretical framework and technical terms mainly from the psychology of programming, it can be made clear what kind of issues are in common with general programming activity and what are special characteristics in computer music programming. Such a contextualization can help clarifying the problems in computer music, to improve the design and the research on programming language design. Furthermore, computer music is likely to be very interesting as a topic in the psychology of programming, as Blackwell describes in [1].

Characteristics of computer music programming seem to be interesting and also beneficial to study on the usability of programming language design. For instance, how to utilize the expertise in music domain for cognitive ergonomics of programming languages is an interesting issue and the nature of creative activity with open-ended goals in computer music programming is also an interesting subject when we consider how programming environments should support exploratory design activity.

5. ACKNOWLEDGEMENT

This work was supported by project grant NRF2007IDM-IDM002-069 from the Interactivity and Digital Media Project Office, Media Development Authority, Singapore.

6. REFERENCES

[1] Blackwell A. and Collin, N. The programming language as a musical instrument, *Proc of PPIG05* (2005)

[2] Burnett, M. et al, End-user software engineering. *Communications of ACM, Vol. 47(9)* (2004)

[3] Carter, J. et al. An Analysis of Interviews with Composers From A Cognitive Styles Perspective. *Proc of ICMC'09*, (2009)

[4] Cascone, K, The Aesthetics of Failure: "Post-Digital" Tendencies in Contemporary Computer Music, *Computer Music Journal*, *Vol. 24(4)* (2000)

[5] Chowning, J. Fifty Years of Computer Music: Ideas of the Past Speak to the Future. *Proc of ICMC'09* (2009)

[6] Collins, N. et al. Live coding in laptop performance, *Organized Sound, Vol. 8 (3)* (2003)

[7] Danneberg D.,The Nyquist Composition Environment: Supporting Textual Programming With A Task-Oriented User Interface, Proc of ICMC'08, (2008)

[8] Détienne, F. Programming Understading and Knowledge Organizaiton, *Cognitive Ergonomics: Understanding, Learning and Designing Human-Computer Interaction*, pp.245-256. (1990)

[9] Détienne, F. Software Design - Cognitive Aspects. *Springer Verlag* (2001)

[10] Green, T.R.G et al. Parsing and Gnisrap *Proc of Empirical Studies of Programmers 2nd Workshop* (1987)

[11] Green, T.R.G and Petre, M. When Visual Programs are Harder to Read than Textual Programs, Proc of ECCE6, (1992)

[12] Halasz F. and Moran T.P. Mental models and problem solving in using a calculator. *Proc of CHI83* (1983)

[13] Hoc J.-M et al. Psychology of Programming, *Academic press* (1990)

[14] Matthews M.V. et al. The Technology of Computer Music. *The MIT Press* (1969)

[15] Mullen, T. Writing Code for Other People: Cognitive Psychology and the fundamental of good software design principle. *Proc of OOPSLA'09* (2009)

[16] Nilson, C. Live coding practice. *Proc of NIME'07* (2007)

[17] Pashler, H., Dual-Task Interference in Simple Tasks: Data and Theory. *Psychological Bulletin Vol. 116* (1994)

[18] Reed D. The use of ill-defined problems for developing problem-solving and empirical skills in CS1. *Journal of Computing Sciences in Collges, Vol.18 (1)* (2002)

[19] Rist, R. Plans in Programming: Definition, Demonstration, Development, *Empirical Studies of Programmers 1st Workshop*, 1986

[20] Shaft, T. and Vessey, I. The role of cognitive fit in the relationship between software comprehension and modification. *MIS Quarterly, Vol.30 (1)* (2006)

[21] Vessey, I. Cognitive fit: A theory-based analysis of the graphs versus tables literature. *Decision Sciences, Vol. 22(2)* (1991)

Seaboard: a new piano keyboard-related interface combining discrete and continuous control

Roland Lamb
Design Products Department
School of Architecture and Design
Royal College of Art
Kensington Gore,
London SW7 2EU
roland.lamb@network.rca.ac.uk

Andrew N. Robertson
Centre for Digital Music,
School of Computer Science and Electronic Engineering,
Queen Mary University of London,
Mile End Road, London, E1 4NS
andrew.robertson@eecs.qmul.ac.uk

ABSTRACT

This paper introduces the Seaboard, a new tangible musical instrument which aims to provide musicians with significant capability to manipulate sound in real-time in a musically intuitive way. It introduces the core design features which make the Seaboard unique, and describes the motivation and rationale behind the design. The fundamental approach to dealing with problems associated with discrete and continuous inputs is summarized.

Keywords

Piano keyboard-related interface, continuous and discrete control, haptic feedback, Human-Computer Interaction (HCI)

1. INTRODUCTION

The Seaboard is a new musical instrument which enables real-time continuous polyphonic control of pitch, amplitude and timbral variation. This novel tangible interface was invented, designed and developed by Roland Lamb, in the context of his studies in the Design Products Department at the Royal College of Art. During the software development stage of the third prototype, Andrew Robertson joined the project to assist with the software design and implementation. The initial motivation for the Seaboard came from the desire to augment the capabilities of the piano and, in particular, to combine the capacity for real-time polyphonic expression with the ability to bend the pitch of every note independently.

Keyboard controllers have been designed with the acoustic piano keyboard as the interface paradigm on which they are based. Many electronic keyboards have pitch wheels which add pitch-bending capabilities. Pitch wheels, however, are of limited use for serious musical performance and do not enable real-time note-by-note polyphonic pitch bending. Piano-like polyphonic pitch-bending interfaces do exist, most notably the Haken Continuum Fingerboard [3], which allows for multiple pitch bends at the same time and also registers the vertical location of an input and its downward pressure. However, the Fingerboard provides the musician with a limited amount of tactile information about finger location, and thus (especially when playing polyphonically) if software correction is not utilized to force tones to snap to the tuning of the twelve-tone scale, then either visual confirmation of each note is necessary or a vibrato technique must be employed. The Rolky Asproyd, designed by Eric Johnstone [5], is a poly-touch controller that makes use of illumination to detect the position of several fingers on a transparent surface, thereby providing control over each note in the chord. Nevertheless, no instrument based on the piano layout has previously provided a musically intuitive way of providing polyphonic pitch-bending capacity while also enabling effective tuned playing.

Figure 1: The Seaboard surface

2. DESIGN

At the broadest level of description, one can identify two ways of making music: traditional musical instruments on the one hand and modular technology—various kinds of synthesizers, sampling, and digital effects—on the other. Traditional instruments provide great depth, refinement, and performative possibilities, but more limited scope, whereas modular technology has enormous scope but is often poorly integrated and difficult to use in real time.

The goal of the Seaboard design process, like that of many new digital interfaces and instruments, has been to deliver an integrated music-creation device which combines the best of both of these approaches. Our aim has been to make use of both the more fundamental intuitive associations (i.e. pressure relates to volume) which enable the learning process and the connection between musician and instrument, and the possibility of taking advantage of more arbitrary

but nevertheless established intuitions in considering key design choices.

In concrete terms, the Seaboard interface takes the basic design layout of the piano keyboard and refashions it with a new surface shape and a new material. The discrete keys of the piano have been physically re-imagined as a single, continuous, non-flat surface, where the relatively raised and recessed areas of the surface correspond with the centers of the white and black keys (See Figure 1). The top of the interface is made of a soft silicone, which rests upon an array of force sensing resistor (FSR) sensors. A software algorithm measures the variations in pressure and location of the pressure peaks in the sensor array, thereby forming a representation of which notes the user is playing on the Seaboard, and sends out the corresponding MIDI or OSC messages.

In addressing the question of how to make an effective tactile instrument that would allow for a wide range of sound and music creation possibilities, and yet remains intuitive, the piano keyboard layout was a good place to start. The visual and logical layout is one of the reasons for the success of the piano, especially as a general interface for musicians to learn basic music theory.

Another reason to adopt the piano keyboard as a starting place lay in its familiarity. New musical instruments and interfaces are often proposed, especially in the digital age, yet comparatively few become established and widely accepted. One reason has to do with the enormous amount of energy that one has to devote to learn a new instrument well, and unless an instrument garners a small community of musicians who play it very well, it is difficult for it to find a path to wider acceptance. Dobrian and Koppelman [**?**] point out that for a new interface to facilitate musical expression, not only must the interface be well designed, for example with respect to mapping gesture to sound parameters, but players must also take the time to master the interface in order to achieve the level of virtuosity we associate with traditional instruments.

In addition to designing the Seaboard in such a way that a musician could transfer keyboard skill and understanding, a strong emphasis was placed on making the new capabilities one that could be learned and endlessly refined through practice, rather than providing easy software workarounds. Highly skilled manipulation of complex sound variables and attributes depends on practice, and the reason practice is effective in these areas is that one can train one's muscular memory to repeat certain delimited tasks without conscious direction or control. We observed that in order for such training to be possible though, there are three requirements: a) the activity must not inherently require visual confirmation and direction (activities that require visual confirmation, like shooting a target, can of course also be practiced, but involve a different form of practice involving hand/eye/body coordination); b) the physical interface must give positional tactile feedback (in the sense that a flat or merely decorated surface does not, and thus some kind of variation in surface, texture, or resiliency can consistent give the user something tactile to which to spatially orient his or her trained automatic muscular adjustments and correction); and c) these physical qualities of the interface have to be standardized and unchanging, so that they provide very similar tactile information in every instance.

3. CONTINUOUS VS DISCRETE

In the development process of the interface, we found it helpful to track the concepts of 'discrete' and 'continuous' through three areas—musical outputs, tactile feedback, and sensor processing. The goal of reimagining the piano keyboard—into a form in which the pitch, volume and timbre of each note could be continuously controlled without a loss of capability with respect to discrete outputs—emerged from a set of assumptions about desirable outputs for a versatile musical instrument.

3.1 Musical outputs

Even if one considers majors areas of music on a spectrum from rhythm, harmony, to melody, we see that conventional musical outputs require discrete, identifiably separate beats or notes, on one side, and more continuous variations in pitch, volume, and timbre on the other side.

We consider a single output a sound with pitch, volume, and timbral characterstics which has a particular duration. Variations in these parameters can either take place continuously within the duration of such an output, or variations can take place between members of a set of discrete outputs. Typically, discrete variations between outputs are more common in rhythmic musical outputs, especially at faster tempos, whereas continuous variation within an output is more common in melodic outputs, especially at slower tempos. To achieve the broadest range of control, one would want to be able to maximize the capacity for discrete variations between outputs and continuous variations within outputs, in terms of pitch, volume, and timbre, and to do so without loss of accuracy.

3.2 Tactile feedback

This aspiration with respect to musical output has to be related to a tactile feedback system which allows one to input both discrete and continuous variations in a way that enable accuracy and real-time micro-adjustments. Specifically, a given range in pitch, volume, and a particular variable that changes some aspect of timbre can be mapped to the x, y, and z axes of a touch-sensitive surface. However, if the surface is flat, then accurately finding the correct locations for discrete or even just starting pitches, for example, is highly problematic.

In the case of the Seaboard, the three-dimensional input surface, made of silicone (see Figure 2), has a wave-shape form where the peaks of the waves produce, when pressed, musical notes corresponding to the notes of a standard musical keyboard. In this way, the Seaboard can, to a significant extent, mimic a conventional keyboard in its operation with respect to enabling the musician to polyphonically play a set of accurate discrete outputs. For example, by pressing on one of the 'peaks' or 'crests' and vibrating a finger, an oscillating signature can be generated by the sensors, which will be interpreted by the processor as a vibrato. In addition, the shape of the surface means that a player can also play into the troughs, i.e. the areas between the crests, to produce microtonal pitches between any half or whole step. Since the input surface is in places continuous, it is able to produce smooth glissando effects on the keyboard.

As shown by Goebl and Palmer [2], tactile information makes an important contribution to the timing accuracy of piano performances. The interface provides three distinct forms of tactile feedback to the user. Firstly, the texture, angle, and other characteristics of the three-dimensional top surface (see Figure 2) give the user immediate information about the location of the touch, in a way that would be impossible on a flat uniform surface where there is no tactile basis for spatial orientation. Secondly, the soft resilient material transmits forces back to the user to provide further tactile feedback to the user who will be able to sense the amount of pressure that he is applying to the interface. Thirdly, the soft material amplifies the variation in the sur-

Figure 2: Closeup of the silicone surface of the Seaboard.

face area of the tactile feedback.

In these ways, the tactile feedback provided by the Seaboard has been designed to maximize the capacity for discrete variations between inputs and continuous variation within inputs, in a way that intuitively matches with the demands of musical outputs.

3.3 Sensor processing

These observations about discrete and continuous musical outputs, and user inputs, relate also to a distinction between two kinds of sensor-processing paradigms, related to the distinction between discrete and continuous touch interfaces found in Hinckley and Sinclair [4], and the discrete and infinite types of sensors described by Vertegaal et al[8].

A given sensor or array of sensors can provide anything from a single binary message to a continuous flow of high resolution data with respect to multiple parameters.

Currently, most user interfaces for the capture of physical movement or touch fall somewhere on a spectrum between two extremes which could be called 'Discrete Control Interfaces (DCI),' which use a set of discrete sensors, which can register either an on or off position to provide simple discrete inputs, and 'Continuous Action Interfaces (CAI),' which register spatial or gestural movement in time to enable more complex inputs based on continuous movement. Ultimately, the spectrum is defined by levels of resolution and numbers of identifiable, distinct parameters, but in practice, especially with respect to pressure based tactile input sensing, the distinction between continuous and distinct is a relevant one.

The DCI side of the spectrum is typified by simple switches and arrays in devices like typing keyboards, and other interfaces that use direct analog (usually switch-based) controls that usually simulate a mechanical action, while the CAI end of the spectrum might be typified by something like a Kinect tracking system that gathers a rich set of data which can then be mapped in various ways. A piano keyboard does measure a continuous action with respect to striking velocity, but is clearly on the DCI side of the spectrum. In the middle of the spectrum we find technologies such as touch screens, touchpads, other two-dimensional touch sensitive interfaces, and devices like a computer mouse, which use a rolling ball or some other continuous action apparatus that allows for continuous input, but might be more limited in terms of the number of parameters that they can track.

The advantages of DCI interfaces are that they allow for clear discrete inputs and they typically form a tactile and rich kinaesthetic input feedback system that does not rely on visual confirmation, since the user can feel a responding pressure when he depresses a key, for example. These advantages relate not just to the kind of sensing device but also to the design of the input surface, the topmost part of the interface with which the user actually interacts. In the case of typing interfaces, the springing quality of a typing keyboard allows the user to understand at the level of kinaesthetic perception that a key has been depressed, and the contours of the individual keys allows the user to make micro-adjustments to facilitate constant, fast, accurate typing without having to look at the keyboard. Indeed, tactile cues have been shown experimentally to strongly affect the accuracy of experts in carrying out touch-typing tasks [7]. For the musician, visual feedback has a greater role during the learning phase than the expert phase, when tactile information about finger location and action and habitual skill play an increasing role in navigating the fingers about the keyboard [8].

The disadvantage of DCIs is that they are limited in the types of input that can be made, especially when the goal is to input quantitative or continuous information, as opposed to qualitatively separate, distinct commands. On the other hand, CAIs have the advantage of allowing for continuous input and subtle or complex forms of information to be communicated very quickly. For example, touch-screen interfaces allow the user to choose between an arrangement of options that can be simultaneously presented in an easily understandable manner.

In the Seaboard design, we found that by using an array of pressure sensors, and then implementing an algorithm which tracked each input we could offer some of the features of both kinds of interfaces. In other words, the non-flat nature of the Seaboard surface, in conjunction with its hybrid sensor-processing paradigm, means that one can choose whether to play a note in a musically discrete or continuous way. Since the Seaboard enables seamless transitions for both discrete input (e.g. inputs to generate the notes of a chromatic scale) and continuous inputs (e.g. glissando and slide effects, timbral and dynamic variations in real time), it is ideally suited for the complexity of both enabling discrete and continuous real-time, note-by note polyphonic variations in pitch, timbre, and volume.

4. PROTOTYPING

The Seaboard has gone through three prototype iterations; the first was a concept non-functioning prototype, the second a small working prototype, and the third a full-size working prototype. Each prototype has allowed us to resolve particular problems and questions that have arisen during the design and development process.

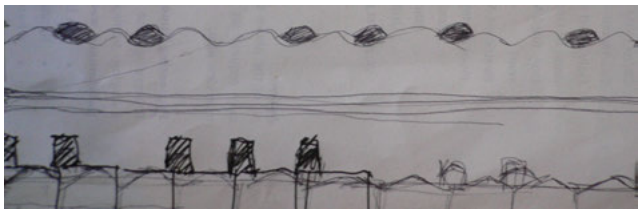


Figure 3: The first sketch of the Seaboard concept.

The first prototype was a concept prototype was based on the sketch shown in Figure 3. The goal was simply to model the main idea in a physical way, and no attempt was made to make it function at that stage. Primarily, then, the Seaboard 1 gave the opportunity to work on the particular shape of the surface, and to test a variety of possible materials. The surface of the Seaboard had to simulate the physical layout of a piano keyboard and the basic size of the distance between each 'wave' was thus given. The trade-off between replicating the exact height differential of the black and white keys on the one hand and making a surface that

was gentle enough in its curvature to allow for easy sliding between positions (and thus pitches) was explored. The relative roundness of the keys, as well as the ways that the surface should extend above and below the key also had to be tested and resolved. In terms of the material, it was necessary to find a solution that had the right level of 'give' and yet also had a fast response time and allowed for a diffusion of forces from the top of the surface through to the sensors underneath. Seaboard 2 was the first working prototype, and thus its development also encompassed the selection of sensors, the electronics to make them work and send correct data, and of course a significant amount of software development. Seaboard 3 has allowed us to develop a more mature prototype and software algorithm, discussed in overview below, and otherwise to resolve all the design questions in a more complete way.

4.1 Input

Marshall and Wanderley [6] find that FSR sensors are the preferred input sensor over linear and rotary potentiometers for relative dynamic control, required for vibrato effects. The input from the Seaboard is via an Arduino Mega multiplexed to provide readings from the array of FSR sensors, each with values ranging from 0 to 1023. The sensor values are currently sampled at approximately 55Hz which provides a relatively low latency when playing.

4.2 Output

The Seaboard sends MIDI information to a sequencer that is used to generate sound. For each note sent, we require the ability to change the volume and pitch of each note, and thus we set up separate MIDI channels for the maximum number of simultaneous notes we wish to send (typically 8). We have made use of Logic and Ableton Live as audio sequencers with which to generate sound from the interface. It is also possible to send OpenSound Control (OSC) messages [9] to communicate the amplitude and pitch of each note.

4.3 Algorithm

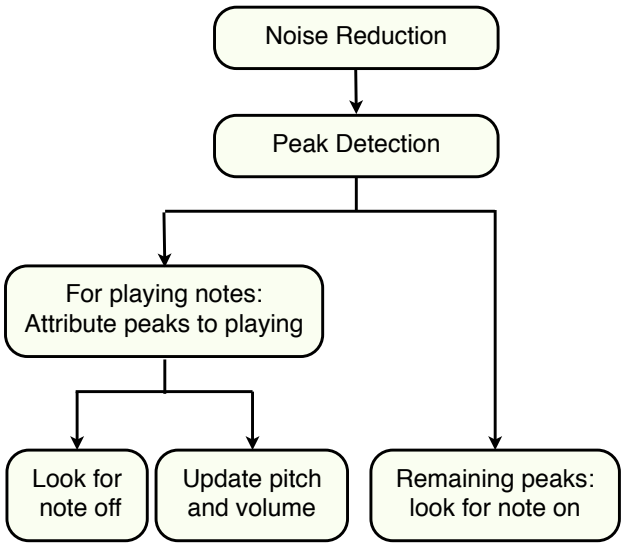


Figure 4: Software Architecture

A diagram showing the design for the software architecture is shown in Figure 4. A noise reduction process makes use of the the maximum sensor values experienced when the instrument is not being played and reduces the sensor values appropriately to prevent false triggering. We then look for peaks in the range of sensors, that is, where a sensor has a pressure value greater than both the adjacent sensors. We calculate a localization in proportion to the pressure that determines each peak's central location and overall pressure. Every MIDI note that is sent out from the Seaboard has an associated location and pressure in terms of the sensor array. Thus, for each playing note, we find the closest peak that has not yet been attributed to an existing note and depending on the pressure, we either update the location and pressure associated with that note or else send a 'Note Off' message and remove the note from our list of playing notes. Then we iterate through any remaining unattributed peaks and if the pressure is greater than a set threshold, we send a 'Note On' message and add the note (with associated peak location and pressure) to the list of playing notes. A mapping function is used to translate between peak location and continuous note location.

5. CONCLUSION

In this paper, we presented the Seaboard, a polyphonic interface that provides continuous dynamic control over the pitch and volume of each note. We have described the iterative design process that led to its construction, highlighting the ethos of the design and the importance of rich kinaesthetic and tactile feedback in new hybrid interfaces that enable both discrete and continuos control.

6. REFERENCES

[1] C. Dobrian and D. Koppelman. The 'E' in NIME: Musical expression with new computer interfaces. *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, 277, 2006.

[2] W. Goebl and C. Palmer. Tactile feedback and timing accuracy in piano performance. *Experimental Brain Research*, 186, 2008.

[3] L. Haken, E. Tellman, and P. Wolfe. An indiscrete keyboard. *Computer Music Journal*, 22(1):30–48, 1998.

[4] K. Hinckley and M. Sinclair. Touch-sensing input devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 223–230, 1999.

[5] E. Johnstone. The Rolky: A poly-touch controller for electronic music. In *Proc. of the International Computer Music Conference*, pages 291–295, 1985.

[6] M. T. Marshall and M. M. Wanderley. Evaluation of sensors as input devices for computer music interfaces. In *Proc. of Computer Music Modeling and Retrieval 2005 Conference, LNCS 3902. Berlin Heidelberg*, pages 130–139. Springer-Verlag, 2006.

[7] E. Rabin and A. M. Gordon. Tactile feedback contributes to consistency of finger movements during typing. *Experimental Brain Research*, 155:362–369, 2004.

[8] R. Vertegaal, T. Ungvary, and M. Kieslinge. Towards a musician's cockpit: Transducers, feedback and musical function. In *Proc. of the International Computer Music Conference*, pages 308–311, 1996.

[9] M. Wright and A. Freed. OpenSound Control: A new protocol for communicating with sound synthesizers. In *in Proceedings of the International Computer Music Conference, Aristotle University, Thessaloniki, Greece*, pages 101 – 104, 1997.

Music Interfaces for Novice Users: Composing Music on a Public Display with Hand Gestures

Gilbert Beyer
University of Munich
Oettingenstr. 67
80538 Munich, Germany
gilbert.beyer@ifi.lmu.de

Max Meier
University of Munich
Oettingenstr. 67
80538 Munich, Germany
max.meier@ifi.lmu.de

ABSTRACT
In this paper we report on a public display where the audience is able to interact not only with visuals, but also with music. The interaction with music in a public setting involves some challenges, such as that passers-by as 'novice users' engage only momentarily with public displays and often don't have any musical knowledge. We present a system that allows users to create harmonic melodies without being in need of a previous training period. Our software solution enables users to control melodies by the interaction, utilizing a novel technique of algorithmic composition based on soft constraints. The proposed algorithm does not generate music randomly, but makes sure that the interactive music is perceived as harmonic at any time. Since a certain amount of control over the music is assigned to the user and to ensure the music can be controlled in an intuitive way, the algorithm further includes preferences derived from user interaction that can be competing with generating a harmonic melody. To test our concept of controlling music, we developed a prototype of a large public display and conducted a user study, exploring how people would control melodies on such a display with hand gestures.

Keywords
Interactive music, public displays, user experience, out-of-home media, algorithmic composition, soft constraints

1. INTRODUCTION
An important goal of interactive public displays reacting to e.g. body movements or hand gestures of passers-by is that interaction has to be such intuitive that novice users can start interacting immediately: Passers-by should be able to walk-up and use the content, or ideally control it in the intended way already by their initial, unconscious interaction. Interactive displays often allow manipulating visual objects that can for example be constituent parts of a brand identity, like a brand logo that can be moved along the display surface by hands or feet. For some reason however acoustic events do not appear at all or play only a secondary role within the interactive experience: often they are delimited to immutable sound objects just supplementing the visual interaction, or statically playing background music. Nevertheless, the enrichment by sound can enhance the interactive experience, and last but not least the identity of many brands that are advertised for on public displays is defined by both a visual and acoustic appearance.

On the other hand, beyond the context of interactive installations in public spaces, interactive music systems have become increasingly popular: with social music games like Guitar Hero, well-known songs can be re-played together, and easy-to-use musical applications for mobile devices such as the iPhone give everyone the possibility for musical expression, even without having any musical knowledge (see Figure 1).

Figure 1. Interactive music making with Guitar Hero and the iPhone

In spite of enjoying great popularity and commercial success, such interactive musical applications have barely been employed in public spaces so far. We propose that the trends of interactive out-of-home media and interactive music making will successfully combine in the future, producing new media that will enable passers-by not only to play with, but also manipulate and shape melodies by means of interactive control mechanisms.

In this work, we present an approach which brings together interactive displays in urban spaces and interactive music systems. When combining public displays and music systems, the question arises how harmonic melodies can be created by 'unskilled' passers-by in a suitable way. With our approach for engaging with music in public spaces, it is possible to create music in many different styles. For example, music can be generated in such a way that it resembles the melody of a well-known song. This way, it is possible to develop interactive musical applications that give musical laypersons the feeling of successfully playing an instrument or composing music.

2. REQUIREMENTS FOR COMPOSING MUSIC ON A PUBLIC DISPLAY
How and if users interact with public displays depends amongst others on the external surroundings and usage context, the type of the display, as well as the number of individuals approaching. Usually passers-by can be assumed to be novice users and laypersons in regard to any application provided on such displays. Especially when it comes to interaction with

music, the question arises how an engaging user experience and a feeling of success can be achieved. Ideally, the demands of the input technique should be simple (yet allow an expressive performance), while the produced music should be as appealing as possible. To give novice users the feeling of successfully composing music in a public space and having fun during their short-term engagement with the application, we follow an approach where they are only capable of manipulating some musical parameters, and a software in the background makes sure that the generated melodies are perceived as harmonic and are reminiscent of some well-known musical themes. The input for the music generation also has to comply with the chosen interaction paradigm, which in the case of public displays is often multi-touch or vision-based interaction with hands. As people stopping in front of public displays are often pairs and small groups of individuals, a public display capable of multi-user interaction should also provide means to play music together in a successful way.

To comply with such requirements, we make use of a novel technique that allows generating music in real-time with respect to so-called preferences that express 'how the music should sound'. With this approach, it is possible to automatically derive preferences from given melodies in such a way that their characteristic properties can be preserved up to a certain extent (e.g. distinctiveness of a melody), while at the same time it is possible to flexibly alter them based on user interaction. Not only can the musical context of a melody be varied (e.g. instrumentation or style), also the melodic material itself can be subject to dynamic changes.

We use three types of preferences: First we use preferences for a single instrument which are derived from user interaction, e.g. a touch display or a motion tracking system. These preferences reflect how the user wants the music to sound, for example 'I want to play fast notes with a high pitch'. In our approach we generate music with only two parameters – 'pitch' and 'energy' – which are usually simple to extract from user interaction with both hands but are also expressive. Intuitively, these parameters continually control the note pitch (high/low) and the speed (fast/slow) at which the instrument should play. Interfaces based on these parameters are easy to play because they require only few musical skills (e.g. making exact rhythmic movements) – nevertheless, they provide much control over the music in a very direct way with immediate musical feedback.

The second type of preferences expresses general melodic rules: With this kind of preferences, it is possible to make the music consistent with a certain musical style (e.g. Hip-hop or Jazz). Furthermore, it is also possible to make the resulting melodies comply with a songs distinct acoustic identity. In most cases, the preferences derived from user interaction will be competing with a songs prominent characteristics, i.e. the user interaction does not fit the tune with respect to both tonality and rhythmics. Since a certain amount of control over the music is assigned to the user, it is inherently not possible to exactly play a given melody note by note. Nevertheless, it is possible to generate melodies which are similar to it by using note pitches as well as tonal and rhythmic patterns appearing in the tune's distinct melody. This way, melodies can be generated considering both interactivity and the recognition of a tune.

At last, we use preferences that coordinate several instruments playing simultaneously, for example a single player with static background music or multiple players among each other. This coordination is made by preferring harmonic intervals between different instruments. Furthermore, it is also possible to coordinate multiple instruments such that they play similar rhythmic patterns.

3. RELATED WORK

Of interest to our work are generally works on user-controllable music within digital media in public spaces. Yet, we currently know of no work that focuses on how to control a distinct melody within the interactive experience. A good overview on algorithmic composition is provided by [3] and [10]. Examples for interactive music composition and generation systems are Electroplankton [8] or Cyber Composer [7].

Related to our work are approaches for imitating musical styles: typical techniques for dealing with this problem are based on musical grammars or statistical models [4]. The Continuator [13] combines style imitation and interactivity. Based on a statistical model, the system is able to learn and generate musical styles e.g. as continuations of a musician's input. Our approach for generating music is based on constraint satisfaction problems. Automatic musical harmonization deals with the problem of creating arrangements from given melodies with respect to certain rules. Pachet and Roy made a detailed survey on musical harmonization with constraints [12].

To our knowledge, there is currently no work describing the combination of music generation and interactive applications in public spaces. In [11] a system for musical performance is described that acquires a user's physical actions and physiological state to alter stored data representing a music piece. In [9] pressure-sensitive controls allow people with disabilities to control the generation of music. The system introduced in [2] uses a performance device to interactively control several aspects of a composition algorithm. A general-purpose position-based controller, where the position signal may also be used for generating music, is described in [14].

Our approach for generating music is based on a reasoning-technique called soft constraints which allows dealing with soft and concurrent problems in an easy way. Bistarelli et al. [1] introduced a very general and abstract theory of soft constraints based on semirings. Building on this work, in [6] monoidal soft constraints were introduced, a soft-constraint formalism particularly well-suited to multi-criteria optimization problems with dynamically changing user preferences. Soft constraints have successfully been applied to problems such as optimizing software-defined radios [15] or orchestrating services [16]. We introduced a soft-constraint based system for music therapy in [5], giving us basic proof of concept with this technique.

4. COMPOSING MUSIC WITH SOFT CONSTRAINTS

To realize interactive, user-controllable music systems in public spaces we developed a technical solution for real-time music generation that helps to coordinate the different characteristics of user interaction, the acoustic identity of a tune and the general harmonic and rhythmic concordance of instruments.

We make use of a framework for algorithmic composition of music which is based on soft constraints [5]. With this framework, music can be interactively generated in real-time by defining preferences as described in the previous section. All preferences can also be generated dynamically, which allows to compose music in real-time, e.g. based on user interaction by continually defining preferences which reflect 'how well the music matches the interaction'. In general, a soft constraint expresses how well an assignment of values to variables (a valuation) matches a desired result. A valuation is a function from variables to values:

$$Valuation = (Variable \rightarrow Value).$$

The extent to which this valuation is desirable can be expressed in various ways. The cited theory introduces a very elegant way

of rating valuations with a set of grades and several operations for combining or comparing grades. Many concrete kinds of grades can be used, for example based on numbers or Boolean values. A soft constraint assigns a grade to each valuation:

$$SoftConstraint = (Valuation \rightarrow Grade).$$

Typically, one is interested in the best possible valuation which can be computed with a general solver for soft constraints. In our application of soft constraints for generating music we want to assign actions to voices: each voice corresponds to a certain sound (e.g. a piano, guitar or synthesizer sound); actions are for example 'play a note' or 'pause'. When an instrument should be polyphonic, it has to have an according number of voices. We use soft constraints to rate action assignments:

$$(Voice \rightarrow Action) \rightarrow Grade.$$

At certain time intervals, each instrument is being asked to state preferences for its own notes. These preferences from all instruments are then extended with global coordination preferences and combined to a single constraint problem. This problem is being solved, yielding an action for each voice which satisfies the preferences best. In the next section, we will introduce a prototype where hand gestures are used to control music. Based on an optical tracking system, we derive two parameters from a user's movements: the total amount of movement (corresponding to the rate of played notes) and the average vertical position of all movements (corresponding to pitch). Based on these two parameters, preferences are generated reflecting the desired speed and pitch. For example, when the user makes fast movements and lifts his hands up, the music should also be fast and have a rather high pitch. Vice-versa, when the user is moving slowly and his hands are down, the music should be slow with a low pitch.

The music should fit the user interaction on the one hand, but we also want it to fit to a given tune on the other hand. This is realized with an additional preference reflecting 'how well the music matches a tune's distinctive melody'. This preference is generated based on a timed transition model representing the tune's note pitches and rhythmic patterns as well as transitions between notes (e.g. 'C is often followed by E or another C'). Our approach is based on a custom transition model which represents sequences of events aligned upon a structured metric grid. Intuitively, the model represents (1) how often an event occurs at a certain metric position and (2) how often other events follow this event at this position. Following typical terms from the closely related area of probability models, the 'events' are called states. The discrete metric positions (representing 'time') are called steps:

$$State$$
$$Step = \{0, \dots, n\}$$

In each step, each state has a certain weight for a given voice. This weight represents how often the state occurs at the given step:

$$stateWeight_{Voice} : Step \times State \rightarrow \mathbb{R}$$

The transitions between states at a given step are represented with the following function. The first two arguments define the original step and state – the third argument defines the next state. Transition weights are always defined for subsequent steps; the state in the third argument is implicitly assumed to be on the next state:

$$transitionWeight_{Voice} : Step \times State \times State \rightarrow \mathbb{R}$$

Figure 2 visualizes a timed transition model with three steps and states. State weights are visualized with black circles: the bigger the circle, the higher the weight. The transition weights are visualized with arrows (a thicker arrow indicates a higher weight). When the model is untrained, all weights are the same. Training the model modifies the weights; the right picture visualizes a trained model with shifted weights.

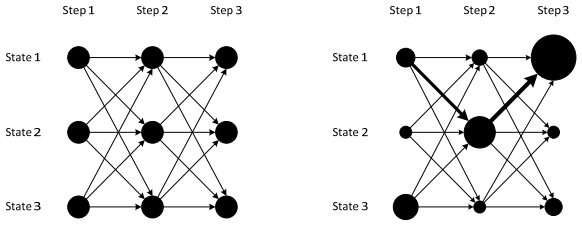


Figure 2. Transition model visualization (left: empty model, right: trained model)

The actual states can be modeled in several ways: the simplest way is to directly use the existing set of actions as states. However, there would be a little disadvantage: if note pitches are directly used within the states, it is not possible to play a model in another tonal scale. If this is desired, it is better to use abstract stages in a tonal scale rather than concrete note pitches. Now, we define a constraint which expresses 'how well an action matches the data represented in the model'. Given the last step and the last actually executed state (the state corresponding to the last action chosen by the constraint solver), we can compute a total weight for each state on the subsequent step. This is done by just summing up the transition weight and the step weight itself:

$$totalWeight_{Voice} : Step \times State \times Step \times State \rightarrow \mathbb{R}$$
$$totalWeight_v(lastStep, lastState, step, state)$$
$$= transitionWeight_v(lastStep, lastState, state)$$
$$+ stateWeight_v(step, state)$$

The constraint itself for a certain voice is constructed based on the last step, the last executed action and the current step. When the sets of actions and states are identical, the constraint can be defined like this:

$$modelConstraint_{Voice} : (Step \times Action \times Step)$$
$$\rightarrow ((Voice \rightarrow Action) \rightarrow \mathbb{R})$$
$$modelConstraint_v(lastStep, lastAction, step)(val)$$
$$= totalWeight_v(lastStep, lastAction, step, val(v))$$

To sum it up, we have preferences based on user interaction as well as preferences reflecting the similarity to a tune, and – in most cases – these preferences will be competing among each other. Furthermore, it is also possible to coordinate several instruments with additional global preferences. In our public display scenario, we define a global constraint which maximizes the amount of musical harmony between the interactive instrument and background music. Soft constraints are very appropriate for dealing with such problems and allow accommodating several concurrent preferences in an easy yet expressive way. When the preferences have been stated, a soft constraint solver can be employed for computing the best possible notes with respect to all preferences. We use a soft constraint solver which was originally prototyped in Maude and that we later implemented in a more efficient version in C#, making it possible to use it in a soft real-time environment.

5. PROTOTYPE AND EVALUATION

To explore how novice users can compose music on a public display with our soft-constraint framework, we developed a prototype with which users can interactively play music with hand movements. The sensing of hands is realized using marker-based techniques. To examine which gestures users would use to manipulate music, we developed several sample applications where the note pitch of the music can be controlled by up-and-down movements of the hands and the rate of played notes by the velocity of hand movements (see Figure 3).

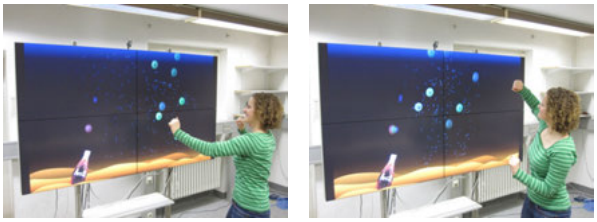


Figure 3. Public display prototype that allows novice users to control music by hand movements

When someone starts to interact with the system, he can realize the connection between his movements and the notes he hears: when the movements become faster, the notes will also play faster – not moving at all leads to silence. The resulting melodies do not only fit to the person's movements, they are furthermore being generated in a way that they comply with a well-known melody. We developed different gesture-based techniques for controlling the music with hands:

The first interaction technique allows the user to control visual elements and acoustic events only with one hand at a time. The note pitch of the music is controlled by up-down movements, and the rate of played notes is controlled by the velocity of movements. The second technique allows the user to control visuals and music with both hands, and for computing note pitch and rate of played notes the mean values of both the hand vertical positions and velocity are taken. The third interaction technique extends the second technique by allowing the user to control the acoustic events (note pitch and rate of played notes) with separate hands, i.e. one hand controls the note pitch and the other hand controls the rate.

Even without any previous instructions, most users were aware that they have control over the music. Only 2 out of 21 people stated they did not recognize the connection between their hand movements and the music. No user stopped interacting while standing in front of the system for a longer period, and the average user made hand gestures for over 90% of the time which gives us further confidence that people understood the basic interaction paradigm. Based on the videos, we analyzed how long it took until people interacted in the way we intended, i.e. when they started to primarily make hand gestures which are relevant for the music generation. The variant based on only one hand took 132 seconds on average, the variant based on the mean values of both hands took 118 seconds and the third variant with separate hands for both parameters took 92 seconds. Even if most users seemed to interact in an effectual way interviews revealed that not everybody did consciously identify the parameters 'pitch' and 'rate of played notes' and how they can be controlled: 12 out of 21 people stated that they used up-and-down movements to control the music and 10 out of 21 people could tell how note pitches can be controlled; only 2 users understood how they can vary the rate of notes. Nevertheless, the results from the user observations make us confident that hand gestures are well-suited for interacting with music without any previous training.

6. CONCLUSION

We introduced an approach for musical composition in public spaces, combining the trends of interactive public displays and interactive music systems in the future. Systems that allow controlling sounds by the interaction can open up new opportunities in advertising, entertainment, or installation art. Yet, as passers-by are usually novice users of any deployed interactive installation and often musical laypersons, we believe that systems where users can play note by note offer fewer opportunities for experiencing music. Instead means should be offered that give users the feeling of success when interacting, while still having a certain amount of control over the music. First user tests with our prototype of a large public display showed that music generation with soft constraints serves this purpose quite well. The next step is to investigate how users interact with the proposed system in the wild.

7. REFERENCES

[1] Bistarelli, S., Montanari, U., Rossi, F. Semiring-based constraint satisfaction and optimization. *Journal of the ACM*, vol. 44(2), 1997, 201–236.

[2] Chadabe, J. *Interactive music composition and performamce system.* US Patent 4526078, 1985.

[3] Essl, K. Algorithmic composition. In: Collins, N., d'Escrivan, J. (eds.) *Cambridge Companion to Electronic Music.* Cambridge University Press, Cambridge, 2007.

[4] Farbood, M., Schoner, B. Analysis and Synthesis of Palestrina-Style Counterpoint Using Markov Chains. In *Proc. of the Intl. Computer Music Conf.* Havana, 2001.

[5] Hölzl, M., Denker, G., Meier, M., Wirsing, M. Constraint-Muse: A Soft-Constraint Based System for Music Therapy. In *Proc. of Third International Conference on Algebra and Coalgebra in Computer Science (CALCO'09).* Springer, Udine, 2009, 423–432.

[6] Hölzl, M., Meier, M., Wirsing, M. Which soft constraints do you prefer? In *Proc. of Workshop on Rewriting Logic and its Applications (WRLA 2008).* Budapest, 2008.

[7] Ip, H., Law, K. Kwong, B. Cyber Composer: Hand Gesture-Driven Intelligent Music Composition and Generation. In *Proc. of 11th International Multimedia Modelling Conf. (MMM'05)*, Melbourne, 2005, 46–52.

[8] Iwai, T., Indies Zero and Nintendo: *Electroplankton.* Game for Nintendo DS, 2005.

[9] Jubran, F. *Sound generating device for use by people with disabilities.* United States Patent 2007/0241918 A1, 2007.

[10] Nierhaus, G. *Algorithmic Composition.* Springer, Heidelberg, 2008.

[11] Nishitani, Y., Ishida, K., Kobayashi, E., Yamaha Corporation. *System of processing music performance for personalized management of and evaluation of sampled data.* United States Patent 7297857 B2, 2007.

[12] Pachet, F., Roy, P. Musical Harmonization with Constraints: A Survey. *Constraints* 6 (1), 2001, 7–19.

[13] Pachet, F. The Continuator: Musical Interaction With Style. In *Proc. of the International Computer Music Conference*, ICMA, Gotheborg, 2002, 211–218.

[14] Wheaton J. A., Wold E., Sutter A. J., Yamaha Corporation. *Position-based controller for electronic musical instrument*. United States Patent 5541358, 1996.

[15] Wirsing, M., Denker, G., Talcott, C., Poggio, A., Briesemeister, L. A Rewriting Logic Framework for Soft Constraints. In *Proc. of Workshop on Rewriting Logic and its Application (WRLA 2006).*Vienna, 2006.

[16] Wirsing, M., Clark, A., Gilmore, S., Hölzl, M., Knapp, A., Koch, N., Schroeder, A. Semantic-Based Development of Service-Oriented Systems. In *Proc. FORTE 2006.* Springer, Heidelberg, 2006, 24–45.

Expanding the role of the instrument

Birgitta Cappelen
Institute Of Design
Oslo School of Architecture & Design
birgitta.cappelen@aho.no

Anders-Petter Andersson
Interactive Sound Design
Kristianstad University
anders@interactivesound.org

ABSTRACT
The traditional *role* of the musical instrument is to be the working tool of the professional musician. On the instrument the musician performs music for the audience to listen to. In this paper we present an interactive installation, where we expand the role of the instrument to motivate *musicking* and co-creation between diverse users. We have made an open installation, where users can perform a variety of actions in several situations. By using the abilities of the computer, we have made an installation, which can be *interpreted* to have *many* roles. It can both be an *instrument*, a *co-musician*, a *communication partner*, a *toy*, a *meeting place* and an *ambient* musical *landscape*. The users can *dynamically shift* between roles, based on their abilities, knowledge and motivation.

Keywords
Role, music instrument, genre, narrative, open, interaction design, musicking, interactive installation, sound art

1. INTRODUCTION
Traditionally an instrument is something a musician plays on to perform music. What the musician plays can be written in advance by a composer, or improvised in the situation by the musician, alone or together with other musicians. In both cases special competence to play the instrument is needed, developed through years of hard training to an amateur or professional level of musicianship. In both cases the *user* is a *musician*, the *artefact* he uses a musical *instrument* and the *action* he performs is *playing*. The role, the artefact and the action are defined *mutually* by the cultural and genre competence the user possesses [5].

Today the computer is used as an instrument in itself, and as part in the construction of other instruments to add new qualities and functions to the instrument. Such computer based instruments have functions lacking in traditional acoustic instruments, e.g. a synthesizer's ability to dynamically filter and modulate the sound signal, and add background accompaniments and beats. All the same, these are, despite their special functionality, instruments to be used by musicians. However, the computer's possibilities can also be used to expand the musical experience and actions for broader groups of users. By computer based instruments we mean both instruments containing electronic hardware like sensors and input devices and software, that are controlled by the musician while playing and based on the programmed rules.

With computer based instruments, people with different musical competencies, can create and experience music together on more equal terms, and in more everyday situations. Music theorists, focusing on the everyday life experience of music, have problematized the mediation [14, 9] and action [23] level of music related activities. With the term "musicking" Christopher Small sees music as a verb, a meaning making activity that includes everyday listening, dancing, creating and performing music. [23] The central is the social activity and experiences, where all present are equal participants, no matter level of expertise or activity. But none of them have treated computer based instruments, and their specific possibilities.

In this paper we show how we have worked with the development in an interactive installation in order to *expand the possible roles*, and "musicking" related *actions*, a computer based instrument can offer. Our aim is to motivate co-creation between different user groups, with different competencies and motivations.

2. ROLES AND ARTEFACTS

What is a Role?
The term *role* originally comes from theatre terminology, but has later been used in disciplines like psychology [19], sociology and within computer games [18]. Role means to play a character in a play, or in social relations. Usually a role is something an actor, or in our case a user, *chooses* or *gets* in a given situation, related to other roles, *situations* or *artefacts*. One can choose which avatar to be in a computer game, or role to play in a social setting. Some roles are given or negotiated in relations to others, like in family settings. Being the oldest son in a family, some things are *expected*, having that role, but other things are *negotiated* in the actual family situation and based on the individual's qualities and history. The roles and related *expectations* are *mutually negotiated* in relation to each other in a specific social and cultural context [19].

Role and Artefact
The role the user chooses, consciously or non-consciously, depends on the *interpretation* the user makes of the situation and artefact. With artefact here we mean any human made object, but our focus is on objects containing computers, ubiquitous computing artefacts.

The interpretation the user does, depend on the user's knowledge, social *belonging*, *context*, and *expectations*. Some thinkers like Martin Heidegger [13], whom has been of huge importance for the Human Computer Interaction (HCI) field, focused on the artefact as tool. Here the *goal* the user wants to achieve by using the tool is the important thing. The artefact affords and enables different forms of interaction. A good tool is for Heidegger something that feels like a part of your body, and the goal becomes to master the tool, or instrument in our case. The qualities of the artefact, tool or instrument determine the user's actions. From this ideal a "good" artefact should be transparent and intuitive [21].

This position has been criticized as being technologically deterministic by sociologist Bruno Latour [16, 17], one of today's most important technological thinkers.

Based on the post-structuralist thinkers from the 1960s like Roland Barthes [6], Julia Kristeva [15], Umberto Eco [10] and Michel Foucault [11], discussing the *role of the reader*, *author* and *text*, constructivist thinkers have gone to the other extreme focusing mainly on the reading and use processes. In media studies this has been an important perspective for the last 20 years, often referred to as reception, consumption or cultural studies. And while media studies increasingly treat interactive media, the media and text theoretical perspectives have become gradually more important for the field of Interaction Design.

The focus in these theories goes from the designer's, composer's or author's history and intention, to the user's competence in the interpretation situation. It is the user's social and cultural competencies that are important for the interpretation of the artefact, or text in the broadest sense. The social and cultural aspects determine the interpretation and meaning of artefacts and the actions they encourage [8].

Artefacts and Actants

Bruno Latour who's studies concern use of physical and technical things [16, 17] has been of great importance for the HCI and Interaction Design field, in particular his Actor Network Theory and theory of mediation [16, 17]. Latour shows how things can act, not only as neutral objects or tools, but as active actors, or *actants*, as he calls them, with abilities to influence scientific results and everyday life.

Shifting Roles

The term *shifting*, like actant, comes from semiotics and originally explains how a reader is motivated by the text to identify with the text's main character. The reader, or in our case the user, can *shift role* from identifying with the main character to a more peripheral character. Latour calls this actorial shifting [17]. The users can also be motivated by the rhetoric of the text, or in our case of the design, to shift position in space to another location and time. By including an old picture of Stockholm the designer can make us imagine being there. Latour calls this *spatial* and *temporal shifting*. What Latour recognised was that when including interaction with physical artefacts, yet another type of shifting takes place, where the user of the artefact not only *thinks about* shifting. Instead the user *delegates meaning* and actions *to the artefact* by *using it*.

Open and Ambiguous

As a part in the earlier mentioned text theoretical discussion from the 1960s the philosopher and semiotician, Umberto Eco, contributed with some very influential texts: "The poetics of the open work" and "The role of the reader" [10]. These texts have been important for the music field because they discuss and analyse works by avant-garde composers like Henry Pousseur and Pierre Boulez. Eco theorises over the poetic, open, interpretative structures of these composers' works. And how the open structure represents *possible music to be realised* by the musicians while performing. From this poetics evolves an ideal of the open and ambiguous work that has been an ideal within all art disciplines, where *time* and *interpretation* are important aesthetic dimensions. And with time comes the interest for narrative and dramatic structure and experience.

HCI based on Heideggerian, functionalistic, engineering ideals has until lately advocated the opposite. Good has been synonymous with disappearing, "natural", intuitive and reduction of ambiguity. But lately, when people with an artistic background has entered the HCI and Interaction Design field the engaging and interpretative potentiality of ambiguity has been introduced to the field [2, 12, 4]. And narratology and dramatology [7, 18] has been an increasingly employed perspective in understanding and designing a use sequence that unfolds over time, especially within computer games [1, 18].

3. THE INSTALLATION ORFI

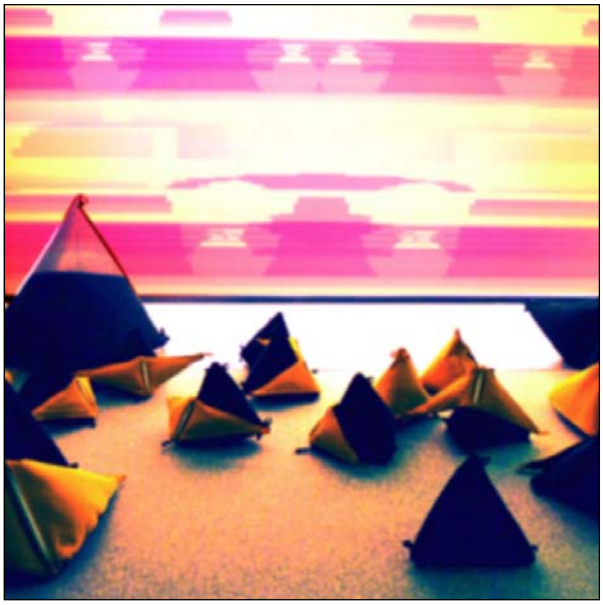


Figure 1. The ORFI landscape, the modules and the dynamic video projection.

Our case in this paper is the interactive installation ORFI. It is a tangible, cross-media installation (see Fig. 1), and a result of over 10 years of explorations within the field of tangible user interfaces for music related activities. Our work is inspired by Eco's thoughts of openness [10], adapted to the field of tangible interaction and directed towards a variety of uses. It is also inspired by Latour's theories of shifting and active actants [17] that take an active role in the communication process, by inviting, provoking and engaging. And thereby staging a real-time realized narrative experience. Knowledge from the field of narratology in relation to interactivity has been a basis for design of the software and content structure. Here Latour's insight in mediation and shifting has been an important framework for our design and composition process.

ORFI consists of 20 tetrahedron shaped soft modules or custom made cushions. The modules are made in black textile and come in three different sizes from 30 to 90 centimetres. Most of the tetrahedron has orange origami shaped "wings" mounted with an orange transparent light stick along one side. The "wings" contain bendable sensors. By interacting with the wings the user creates changes in light, video and music. Two orange tetrahedrons contain microphones. ORFI is shaped as a hybrid, a hybrid between furniture, an instrument and a toy, in order to motivate different interpretations and forms of interaction. One can sit down in it as in a chair or play on it as on an instrument, with immediate response to interaction. Or one can talk, sing and play with it, as with a friend and a co-musician in a communicative way, where ORFI answers vary musically after some time.

Every module contains a micro computer and a radio device, so they can communicate wireless with each other. The modules can be connected together in a Lego-like manner into large interactive landscapes. Or, the modules can be spread out in a radius of 100 meters. So one can interact with each other sitting close, or far away from each other. There is no central point in the installation, it is like a field [8]. The users can look

at each other or at the dynamic video they create together. Or one can just chill out and feel the vibrations from the music sitting in the largest modules as an immersive, ambient, experience.

The installation has a 4-channel sound system that makes listening a distributed experience. ORFI consists of several music genres, which the user can change between. Some of the genres use sound files that can be combined, following musical principles for layering and sequential ordering. In other genres the music and the dynamic graphics is based on programming code, making it possible to order content in layers and sequentially, based on how the users interact. These rules for interaction and music composition have been described in detail in earlier publications. [8, 3]

The many possibilities, like mobile modules and many genres to choose and negotiate between, reflect our goal to facilitate communication between different users and situations.

4. OBSERVATION AND DISCUSSION

The ORFI installation has been evaluated and user tested in many ways, and on different stages throughout the design process. After finishing the installation we have done several sessions of user observations in a usability lab with families and other user constellations.

Five families, with disabled children, spent between one and two hours at our "home look-alike" usability lab, while we were sitting behind a glass walls observing and filming from 4 angles, recording video material for later analysis. After the test period we made in-depth interviews with all family members present. We also made additional user testing at a hospital rehabilitation centre where patients made weekly visits at a Multi Sensory Environment. Here 12 users experienced ORFI for one hour, twice, with a week in between. The observations were recorded, with two fixed and one motor-controllable video camera. Together with the therapists we moved the camera during sessions and watched what were happening on a TV screen from a neighbouring room. Before the session we had introduced the therapists to ORFI on a technical level. All users were brought by their care person or family member, and they spent the hour together in the room.

4.1 One family in ORFI

In this paper we have chosen to present observations and analysis of only one family. The reason is that this family is representative for our findings in relation to ***taking roles*** and ***shifting roles***. During only one session in the usability lab we observed how they used ORFI in a *multiple of ways*: as an instrument, a co-musician, a communication partner, a toy, a meeting place and an ambient musical landscape.

The family consists of six members: mother, father and four children. The youngest boy, with multiple disabilities, was 6 years old when we did the observations. He had two older brothers, age 8 and 11, and a teenage sister.

The observation of this family shows the relational potentiality of an open design like ORFI. This because ORFI offers many people to be present and share the musicking experience on their own terms, by offering people a possibility to take roles and shift between many roles.

Mother and Son – from Instrument, Communication Partner, to Co-musician

Mother and 6 year old son sat down on the floor, facing each other. She reached for one ORFI module, in order to see how it worked. She turned it over and squeezed its' wings to understand the causal relation between her actions and the responses. She tried to *master* ORFI and thereby gave it the role of a tool or instrument. The son, on the other hand, watched the mother's tryouts and listened to the sound. Accordingly, he took the role of the *listener and spectator without interacting*. ORFI responded with a short light and sound to each of the mother's interactions. ORFI created a *stable, non-shifting,* response to the mother's repeated interactions with the same module. ORFI became an instrument that always gave the *same response*.

Role shifts during breaks. The mother continued to interact with one module and made a short hesitation, a break between each interaction. She repeated it three times. ORFI registered the repetitions of interaction-break, and after the third time, answered with *shifted, delayed response* in sound, in addition to direct response in light and graphics. The direct light response synchronised with the mother squeezing the wing as opposed to the sound response when she released the wing. Mother and son smiled and looked at each other. The mother and son had shifted focus from expecting a response to the initial action to focus on the break, the interval in between the sounds and the actions. They had also shifted from treating ORFI like an instrument, to treating it as a *communication partner*. That role was strengthened through *imitation* and *variation* as the mother kept on interacting. As she continued to persist on interacting with a break, ORFI increased the number of shifted responses until they formed a sequence of sounds on every release. The son shifted role from listening to *communicating* through *smiles* and *glances*. ORFI shifted role from mechanical instrument to *communication partner*.

Co-musicians create to the beat. The mother chose a particular module that played a rhythmical beat, continuously like a background drummer. She interacted with the wing on another module and started to synchronise her movements, so that they followed the beat. The son imitated her actions and moved his head and arms to the beat as if he was dancing. ORFI registered the mother's degree of synchronisation to the beat. If she was on the beat, off-beat or out of beat. If she managed to synchronise many times in a row, and over longer time. When she succeeded to synchronise three times, ORFI responded with motifs and riffs with rhythmic, melodic, timbre and chord shifting *variations*, within the musical genre. These shifting responses were played by ORFI, in addition to the direct response. ORFI took the role of *co-musician*, making musical imitations and variations, as would a *member of a band* when playing music together with another band member. The mother shifted her role from communication partner to *co-musician*, shifting down to the rhythm, trying to *synchronise her actions to the music*, expecting more variations from ORFI. The son shifted role to a *co-musician*, musicking and interpreting the sound as music through *dance movements*.

Daughter –Instrument to Meeting Place

The teenage daughter entered the room and moved towards the corner, creating her own space, away from the mother and brother. The daughter lifted up and interacted with modules from the floor, one after the other. She squeezed their wings, and saw what happened on the video projection. Each time the modules answered in light, sound and graphical changes in the video. Just as the mother, she took the role of a person *trying to master* ORFI. Each time ORFI registered and interpreted her separate interactions and answered back directly. On answering directly ORFI took the role of an *instrument*.

The teenage daughter discovered small hooks in each corner of the triangular module. She connected two modules with rubber bands that she found next to them, and put them back on the floor. She looked at them and continued to connect and try out different combinations, until the modules created an arm chair. She sat down. In the course of investigating hooks and rubber bands, she gradually had shifted role from trying to master, to *co-create* her own furniture. With its open and

modular design of hooks and rubber bands ORFI contributed and afforded the daughter's creation. ORFI shifted role from instrument to *meeting place* with references to furniture and teenage room.

Father – Ambient Soundscape

Meanwhile, the father entered the room and sat down in one of the largest, black ORFI modules. The module had speakers and played music created by the users' interaction, which made it an *ambient soundscape*. Other activities in the room, became a background to his relaxing activities.

Brothers – from Instrument to Partners

The two older brothers took one ORFI module each and started to bend the wings. They recognised the direct response and tried to *master* ORFI as an *instrument*. They started to tease each other, punching the other with the modules. It developed into full pillow war. ORFI registered their intense overlapping actions. After three times, ORFI gave a shifting response with *harsh timbre*. It ended with the younger brother covering the older with modules. They shifted roles from trying to master ORFI to *tease*, *compete* and *negotiate* their actions as *communication partners*. ORFI also took the role of a communication partner when it gave shifted and delayed response, with harsh timbre, *imitating* the teasing actions. As the big brothers laughed out loud, the little brother looked at them with admiration. The mother smiled and the teenager sighed. The father experienced their activities as ambient vibrations. ORFI became a *meeting place* for the whole family, where everyone could musicking on their on terms, at the same time, and still experience companionship [22] in the family.

5. CONCLUSION

In this paper we have presented and argued how to design a musical interface to facilitate *musical co-creation* between *diverse users*, by offering the users possibilities to *shift roles dynamically*. By designing an open interactive installation that offered the users possibilities to *take* and *shift* between many roles, the *musicking* experience was enriched. Instead of just being a *performer* that mastered an *instrument* or a passive listener, the user was able to shift between many roles; from being a musician playing on an instrument, to a co-musician playing intense with another *co-musician*. Or from being a communication partner communicating with a *friend*, to a more passive user, who just experienced an ambient, tactile, musical *landscape*. We observed that the users and the artefacts continuously and mutually *negotiated* the *roles*, and the *relevant actions* to expect and perform. And *how* the actions were performed and responded to. One may *hit* an instrument intensely, and one might *throw* a pillow. But one *listens* to a friend or a jazz co-musician, before one *answers*. The negotiation of roles is based on the user's cultural and social *genre* competence.

Inspired by Eco's ideal of *open wor*ks [10], Small's term *musicking* [23] and Latour's theories of *actants*, *mediation* and *shifting* [16, 17], we designed an open installation, ORFI, which we have presented in this paper. We call it an open field, because of its openness to many interpretations, interaction forms and roles to take. We have argued for the constantly accessible possibilities of shifting roles in order to co-create the musicking experience. These possibilities opened up for the user to participate in the musicking on his own terms and in his own manner. So instead of interacting in the same way, people with different abilities and competences, can all interact in their own manner and level of activity. By using the abilities of the computer we have shown and argued how we have expanded the role of the instrument, in order to facilitate musical co-creation between a diversity of users.

6. ACKNOWLEDGEMENTS

We like to thank Fredrik Olofsson for all his work with music, software and graphics, and the Research Council of Norway's VERDIKT programme that makes it possible to continue the research in the RHYME-project (www.RHYME.no).

7. REFERENCES

[1] Aarseth, E. *Cybertext : perspectives on ergodic literature*. Bergen : Univ. of Bergen, 1995.

[2] Andersson, A-P. Cappelen B. Ambiguity—a User Quality, Collaborative Narrative in a Multimodal User Interface. *Proc. AAAI, Smart Graphics*, Stanford. 2000.

[3] Andersson, A-P. Cappelen, B. Same But Different, Composing for Interactivity, *Proc. AudioMostly08*, Luleå Univ., Interactive Institute, 2008, 80—85.

[4] Aoki, P & Woodruff, A. Making space for stories: ambiguity in the design of personal communication systems. *Proc. CHI'05*. 2005, 181-190.

[5] Appadurai, A. *The Social Life of Things: Commodities in Cultural Perspective*. New York: Cambridge Univ. 1986.

[6] Barthes, B. Text The Death of the Author. *Image, Music, Text*. Fontana: London1977/67.

[7] Bell, M.S. *Narrative Design, a Writers Guide to Structure*. Norton, New York, 1997.

[8] Cappelen, B. & Andersson, A-P. From Designing Objects to Designing Fields - From Control to Freedom. *Digital Creativity 14(2).* 2003, 74—90.

[9] DeNora, T. *Music in Everyday Life*. Cambridge University Press. Cambridge, 2000.

[10] Eco, U. *The role of the reader : explorations in the semiotics of texts.* Bloomington : Indiana U.P. 1979.

[11] Foucault, M. What is an Author? *Language, Counter-Memory, Practice*. Ithaca, New York: Cornell University Press, 1977, 124-127.

[12] Gaver, W., Beaver, J., Benford, S. Ambiguity as a resource for design. *Proc. CHI'03*. 2003.

[13] Heidegger, M. *Being and Time* (Sein und Zeit). New York : HarperPerennial/Modern Thought. 2008/27.

[14] Hennion, A. Music and Mediation, Toward a New Sociology of Music. *The Cultural Study of Music, a Critical Introduction*. Clayton, M., Herbert, T., Middleton, R. (ed.). Routledge. New York. 2003, 80—91.

[15] Kristeva, J. From symbol to sign. *The Kristeva Reader*. Oxford : Blackwell, 1986, 74-88.

[16] Latour, B. *Aramis or the Love of Technology*. Cambridge, Mass. Harvard Univ. Press. 1996.

[17] Latour, B. *Pandora's Hope : Essays on the Reality of Science Studies*. Cambridge Mass. Harvard Univ. Press. 1999.

[18] Laurel, B. *Computers as Theatre*. Addison-Wesley. Reading Mass. 1993/1991.

[19] Mead, G.H., Morris, C.W, *Mind, Self, and Society*. University of Chicago Press, *1934*/1972.

[20] Musical Fields Forever: www.musicalfieldsforever.com, visited, April 25 2011.

[21] Norman, D. *The Design of Everyday Things*. London : MIT, 1998.

[22] Ruud, E. Musikk gir helse. Aasgaard, T. (ed.), *Musikk og helse*, Cappelen Akademisk Forlag. Oslo. 2006.

[23] Small, C. *Musicking : the meanings of performing and listening*. Wesleyan University Press. Connecticut. 1998.

.

Wireless Digital/Analog Sensors for Music and Dance Performances

Todor Todoroff
Institut Numediart - UMons
Bd. Dolez 31
7000 Mons, Belgium
todor.todoroff@skynet.be

ABSTRACT

We developed very small and light sensors, each equipped with 3-axes accelerometers, magnetometers and gyroscopes. Those MARG (Magnetic, Angular Rate, and Gravity) sensors allow for a drift-free attitude computation which in turn leads to the possibility of recovering the skeleton of body parts that are of interest for the performance, improving the results of gesture recognition and allowing to get relative position between the extremities of the limbs and the torso of the performer. This opens new possibilities in terms of mapping. We kept our previous approach developed at ARTeM [2]: wireless from the body to the host computer, but wired through a 4-wire digital bus on the body. By relieving the need for a transmitter on each sensing node, we could built very light and flat sensor nodes that can be made invisible under the clothes. Smaller sensors, coupled with flexible wires on the body, give more freedom of movement to dancers despite the need for cables on the body. And as the weight of each sensor node, box included, is only 5 grams (Figure 1), they can also be put on the upper and lower arm and hand of a violin or viola player, to retrieve the skeleton from the torso to the hand, without adding any weight that would disturb the performer. We used those sensors in several performances with a dancing viola player and in one where she was simultaneously controlling gas flames interactively. We are currently applying them to other types of musical performances.

Keywords

wireless MARG sensors

1. INTRODUCTION

There is growing need for improved devices to track the gestures and movements of musical performers and dancers on-stage, for various kinds of interactive performances. Systems that require only a light and fast setup, that are robust enough to take on tour and that don't modify the dancer's appearance. This often excludes the use of some well established technologies used in motion pictures or in the gaming industry, like putting visible markers on the body and using a large array of cameras, or demanding the dancer to wear a special and cumbersome suit fitted with arrays of sensors. And the price should remain affordable for artistic projects.

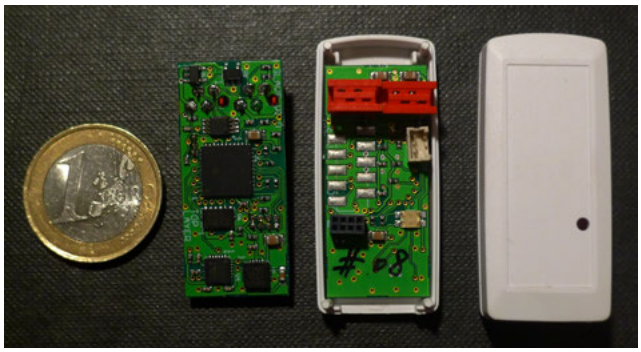


Figure 1: *The sensor nodes, from left to right: 1 Euro coin, top PCB view, bottom PCB view in the lower part of the box, showing red connectors for I2C Bus and power supply (empty pads can be used to solder up to 6 additional analog inputs directly on the PCB or with a micromatch connector), and box closed, with hole to see the bicolor LED.*

This design started as a part of a wider project to build hardware and software tools to enable interactive performances whereby a dancer controls music and fire in the form of software-controlled gas flame projectors. We also developed software for a stereoscopic camera to follow the dancer in a difficult environment with flames. We will come back to this in the application part of the paper.

When we started the project for the fire control in 2009, there were no affordable sensors on the market fulfilling our needs and we decided to build a new system, using the latest available sensing chips and low power wireless technologies to improve our previous designs, extending the capacities by combining 3-axes accelerometers, magnetometers and gyroscopes while reducing the size significantly.

We plan to make the sensors available commercially with a Max/MSP toolbox to communicate with the sensors, decode and analyze their signals. It takes care of bi-directional communication between the sensors and Max, allowing the user to tailor the sensor system to his needs and giving him tools to define his sensor name space. The received data is decoded and scaled and the value of each sensing axis is available using a simple Max receive object in physical units: g, Gauss and deg/s. Attitude information is given for each node in quaternion representation. The toolbox will include improved versions of the tools we developed for the *Dancing Viola* project [21]: hit detection, DTW-based gesture recognition [3] and mapping by interpolation[22].

2. SENSORS

All commercial wireless sensor interfaces designed for artists (Eowave [6] Eobody2 HF, Interface-Z [11] Wiwi or Mini-HF, Infusion System [10] I-cubeX, La Kitchen [13] Kroonde, ...)

have only analog inputs, limited to 16 channels and 10 or 12 bits ADCs. While they allow users to connect various sensors without any additional programming, they are all quite limiting in terms of the number of available channels and they imposes a heavy harness of wires. Indeed, as we wanted to fit each sensor node with 3-axes accelerometer, magnetometer, gyroscope and temperature sensor for calibration, it meant 10 channels plus ground and power, or 12 wires per sensor node. And we wanted a system capable of sampling 3 to 6 sensor nodes on a dancer at 100 Hz.

A system with sensor nodes of similar capacities had been described in [7], though with a different approach. There is an obvious trade-off between that system, truly wireless, even on the body, but with bigger nodes as each one carries its own emitter and battery, and our system with sensors connected through a digital bus on the body. We believe that our approach, with very flat sensors, invisible under the clothes, offers more freedom of movement to the dancer, particularly for movements on the ground, despite the need for cables on the body. Light-weight sensors have the additional advantage of having a small inertia than heavier sensors, which allows them to follow more closely the movements of the limbs of the dancer they are attached to.

Following the experience of the sensor system developed in 2006 at ARTeM [2] for the *Quartet Project* [16], *De deux points de vue* [5] and *Dancing Viola* [21, 4], we kept a master/sensor nodes architecture while reducing the form factor and adding sensing capabilities. They communicate through a 4 wire 400kHz I2C bus on the body: a bidirectional data (*SDA*) and a clock (*SCL*) link, a common ground (*GND*) and a power supply line (*VDD*). The global architecture is shown in Figure 2.

2.1 Sensor chips choices

We made an extensive search at the end of 2009 for our first prototype. There were obvious choices for 3-axes digital magnetometers (Honeywell [9] HMC5843) and accelerometers (STMicroelectronics [19] LIS302DLH or the Analog Device [1] ADXL345). We chose for the later both for its wider range, keeping a constant resolution of 4mg/LSB at all ranges, and for its additional functions. But 3-axes gyroscopes were not yet available and we had to chose a combination of an x/y-axes gyroscope and a z-axis one. Because of the amount of external components needed, we chose for the InvenSense [12] IDG-650 and ISZ-650 rather than for the STMicroelectronics [19] LPR550AL and LY550ALH gyroscopes. We used a PIC18F2423 for its 12-bit DACs, adding four times oversampling for better precision.

In our latest design, in 2010, we use the newly available Invensense ITG-3200 digital 3-axes Gyroscope and added 6 channels of ADC for optional additional sensors (pressure, flexion, light, ...), all on a 17x38 mm PCB that fits into a tiny USB key box (Figure 1). The boxes are 10 mm thick and if even flatter sensors are needed, it is possible to remove the connectors altogether, to solder the 4 I2C Bus and power wires directly on the PCB, and to enclose the sensor in resin, reaching less than 4mm thickness.

2.2 Wireless sensor system architecture

We tested several low power wireless transmission technologies to see how far we could reduce the size and weight of the battery: ZigBee, SimpliciTI, Bluetooth. But we found huge disparites between the announced data rates and the measured ones: ZigBee and SimpliciTI could not be used reliably with more than one or two sensor node. BlueTooth could handle three nodes at 100Hz, but only 50Hz gave a decent latency, as packet sizes increased with data rate. While a low power WiFi module [17] had no problem

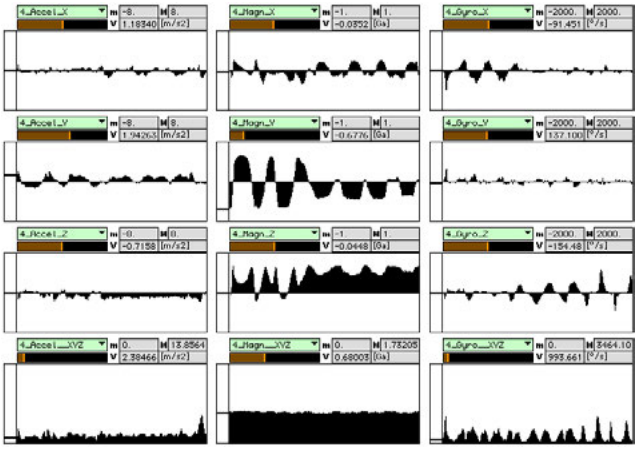


Figure 3: *Max/MSP Display of the 3 axes of the Accelerometer, magnetometer and gyroscope and total amplitude of each for one sensor node.*

transmitting eight sensors nodes at 100Hz, with the added benefit of a smaller and constant latency thanks to the use of a match character to send data in a single IP packet. Despite higher transmission power, as the transmission time is reduced thanks to the high throughput, the average WiFi power consumption was similar to BlueTooth and was chosen as shown in Figure 2.

2.3 Max Toolbox

Contrarily to most commercial systems, bi-directionnal communication allows the user to remotely and dynamically set up, directly from Max/MSP, the sampling period, which of the on-board sensors need to be transmitted by each node, including the number of ADC channels, allowing the user to tailor his system and to optimize bandwidth. Unused on-board sensors can be put to sleep in order to economize power. Various configuration parameters of the accelerometers, magnetometers or gyroscopes can also be modified in real-time: their range, their individual sampling frequency, the cut-off frequency of their low-pass filter, self-test of the accelerometer, degaussing the magnetometer, etc.

The received data is decoded by an external Max object. The user can define a name space for each sensor. The values are then scaled depending on gains and offsets. Those are either given by the user or automatically computed for the accelerometers and the magnetometers within the Max external after the user records the data in 6 different positions. A function to zero the offsets of the gyroscopes is also provided. And the value of each axis is made available using a simple Max receive object in meaningful units: g for the accelerometer, Gauss for the magnetometer and deg/s for the gyroscope. They can be displayed as in Figure 3.

2.4 Attitude computation and skeleton

MARG (Magnetic, Angular Rate, and Gravity) sensors allow for a drift-free attitude computation using Kalman filters with a quaternion representation of the angles [15] in order to avoid singularities associated to Euler angles. But we integrated in our external Max object a method by Madgwick [14] that gives good results even at low sampling rates. At 100Hz, using dynamic values of gains β and ζ to avoid disturbing the quaternion computation when the total acceleration diverges from 1g, we obtain excellent results, even when shaking the sensors or performing hits. The method performs quite well even at 50Hz.

As the amount of sensor nodes we can connect to a master

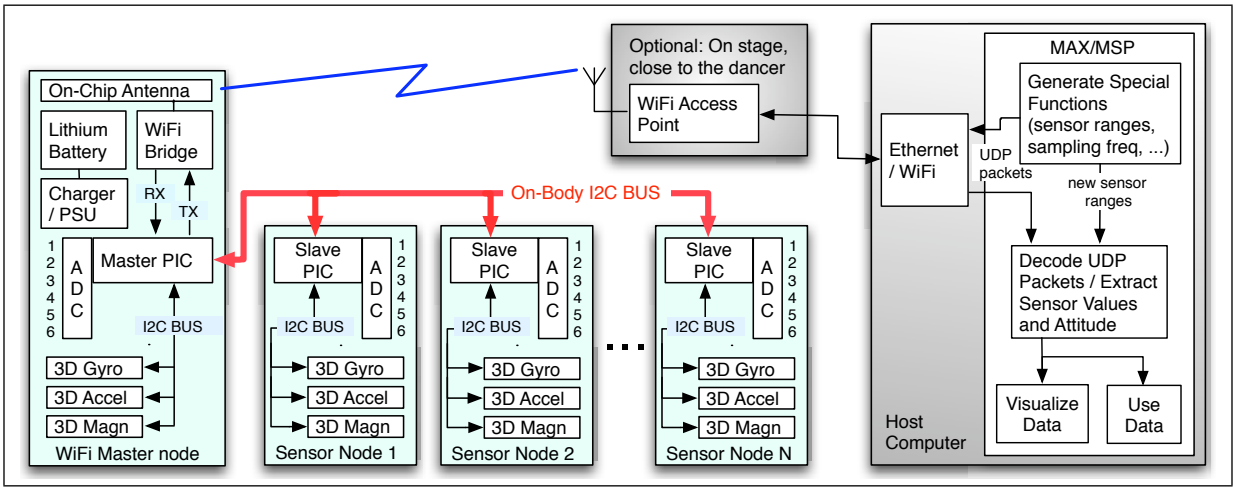


Figure 2: *Wireless sensor system global architecture, with all the bi-directionnal transmission paths (network, wireless, serial, local and on-body I2C) and analog inputs.*

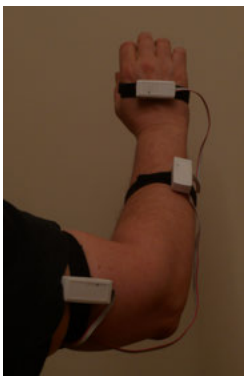


Figure 4: *Three sensor nodes attached to the arm and hand using velcro strips.*

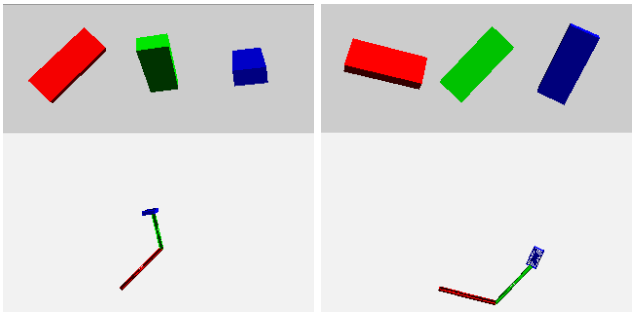


Figure 5: *Attitude of the upper arm (red), lower arm (green) and hand (blue) and reconstructed skeleton in Jitter.*

is limited by the bandwidth of the on-body I2C Bus, halving the sampling frequency allows to double the amount of nodes. Tests showed that our system could sample 3 sensors (master included) at 200 Hz, 8 sensors at 100 Hz and we may extrapolate to at least 16 sensors at 50Hz (we are waiting for a new batch of sensor nodes to get the real value).

If enough sensors are placed on a limb, for instance upper and lower arm plus hand as in Figure 4, a skeleton can be animated in jitter and the position of the hand in regard to the shoulder can be computed (Figure 5), We can thus get the skeleton of upper body at 100 Hz with 8 sensors or the complete body skeleton at 50 Hz with 16 sensors.

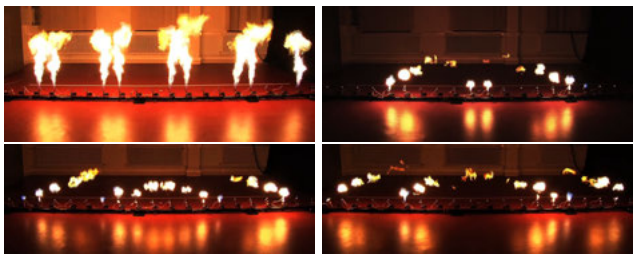


Figure 6: *On the fire ramp, each flame is individually controlled and short bursts of gas can generate fire balls.*

3. APPLICATION: CONTROLLING MUSIC AND FIRE

The sensors were used on the project *FireTraSe* [8] with pyrotechnician Pierre D'haenens who built a patent pending ramp of 20 flames projectors at ShowFlamme [18].

The height of each flame of the ramp can be independently controlled by software and we designed several pattern generators driven by the combination of the sensors signals, video analysis with a stereoscopic camera and sound analysis, using the mapping scheme described in [21]. The height of a gas flame depends on the opening of the corresponding valve, the amount of time that valve is kept open and the upstream gas pressure. Leaving the valve opened for a sufficient amount of time at a specific value will generate a flame of a corresponding specific height. If we modulate the opening of the valve in time, we can as well generate fire balls as shown on Figure 6. This phenomenon depends on the inertia and *time to live* of the projected material. We programmed pattern generators to generate sets of 20 control values varying over time. Though designed to control gas valves to produce flames, the software could drive any number of valves controlling any fluid, like smoke or water.

In that framework we also developed in our lab video analysis tools for a stereoscopic camera in order to track reliably the gestures, the position and height of a dancer despite the presence of the flames [20]. In short, we use the distance information from a Stereo-on-Chip Videre Design camera [23] to remove the image of the ground so that changes of light and shadows generated by the flame patterns don't interfere with the blob detection. Something that could not have been done with an IR camera or background subtraction techniques. The camera can be placed

Figure 7: *Dominica Eyckmans dancing while playing viola.*

in any position as we perform a coordinates transformation from pixel position and depth to stage coordinates (x,y,z).

As the camera reconstructs everything it sees, we define planes in front of the walls, the fire and over the ground. They serve as thresholds to suppress unwanted information, leaving only the dancer's 3D reconstruction. Blob tracking gives us a bounding box in stage coordinates, providing the (x,y) centre of the performer and his height.

The whole system worked within Max/MSP/Jitter, except for the video tracking, running on a separate Linux computer communicating through OSC.

In combination with the tools developed for the *Dancing Viola* project, we blended the three modalities: position tracking, gestures analysis and sound analysis. Figure 7 shows the performer playing the viola, controlling flames and sound transformations of her acoustical instruments as well as triggering and modulating pre-recorded sounds. The sensors were placed on her legs and torso. The attitude extraction, the interpolation tools [22] and the DTW gesture recognition [3] can be combined to give increased control.

4. CONCLUSIONS

We believe our sensors system is an improvement in size, capabilities and resolution over other systems in the same price range. The combination of high resolution digital 3-axes accelerometers, magnetometers and gyroscopes allows for a robust attitude computation and the choice of WiFi allows several performers to share the same wireless channel. Attitude extraction and skeleton reconstruction provide data that improves significantly DTW gesture recognition. Indeed, gestures measured only with accelerometers and gyroscope lack drift-free horizontal plane orientation that might help discriminate between different gestures. Another issue with acceleration and angular speed data is that they do change of value when a movement is performed faster or slower, inducing an increase of the DTW error when the execution speed diverges from the recorded reference gesture. Attitudes and positions don't suffer from that problem and are therefore more suitable for DTW.

We are working on better visualization tools to combine the attitude of each node, with the display of the smoothed and maximum values of the individual sensing axes, devising appropriate representations for accelerations, angular speeds and magnetic field. And we are investigating percussionist gestures, taking into account the preparation gesture before the hit to determine the sound being played.

5. ACKNOWLEDGMENTS

We would like to thank D. Binon at SEMI for his help in designing the PCB and soldering the first sensor prototypes as well as O. Schevens, D. Lekime and J.-Y. Parfait at Multitel for their help in designing the final version.

Research supported by `Numediart`, a long-term research program centered on Digital Media Arts, funded by the Région Wallonne, Belgium (grant N°716631).

6. REFERENCES

[1] Analog Devices. http://www.analog.com/.

[2] ARTeM - Art, Recherche, Technologie et Musique. Brussels, Belgium. http://www.artem.be/.

[3] F. Bettens and T. Todoroff. Real-time dtw-based gesture recognition external object for max/msp and puredata. In *Proc. SMC '09*, pages 30–35, 2009.

[4] Dances with Viola. http://www.danceswithviola.org/.

[5] De Deux Points de Vue. http://www.michele-noiret.be/index.php?page=de-deux-points-de-vue.

[6] Eowave. http://www.eowave.com/.

[7] M. Fernström. Celeritas: Wearable wireless system. In *Proc. NIME '07*, pages 205–208, 2007.

[8] FireTraSe. http://www.numediart.org/projects/project-09-3-firetrase/.

[9] Honeywell. http://www.honeywell.com.

[10] Infusion Systems. http://infusionsystems.com/.

[11] Interface-Z. http://www.interface-z.com/.

[12] InvenSense. http://invensense.com/.

[13] La Kitchen. http://www.la-kitchen.fr/.

[14] S. O. H. Madgwick. An efficient orientation filter for inertial and inertial / magnetic sensor arrays. 2010.

[15] J. L. Marins, X. Yun, E. R. Bachmann, R. McGhee, and M. J. Zyda. An extended kalman filter for quaternion-based orientation estimation using marg sensors. pages 2003–2011, 2001.

[16] Quartet. http://www.quartetproject.unsited.org/.

[17] Roving Networks. Rn-131: Wifly gsx 802.11 b/g wireless lan module. http://www.rovingnetworks.com/wifly-gsx.php.

[18] ShowFlamme. http://www.showflamme.be/.

[19] STMicroelectronics. http://www.st.com/.

[20] T. Todoroff, R. Benmadhkour, and R. Chessini Bose. Multimodal control of music and fire patterns. In *Proc. ICMC '11*, Huddersfield, England, 2011.

[21] T. Todoroff, F. Bettens, W.-Y. Chu, and L. Reboursière. Extension du corps sonore - dancing viola. In *Proc. NIME '09*, pages 141–146, Pittsburgh, Pennsylvania, USA, 2009.

[22] T. Todoroff and L. Reboursière. 1-d, 2-d and 3-d interpolation tools for max/msp/jitter. In *Proc. ICMC '09*, pages 447–450, Montreal, Quebec, Canada, 2009.

[23] Videre design. http://www.videredesign.com.

Real-time control and creative convolution

Exchanging techniques between distinct genres

Trond Engum
Music technology
NTNU, Department of music
7049 –N Trondheim
(+47 73590092)
trond.engum@ntnu.no

ABSTRACT

This paper covers and also describes an ongoing research project focusing on new artistic possibilities by exchanging music technological methods and techniques between two distinct musical genres.

Through my background as a guitarist and composer in an experimental metal band I have experienced a vast development in music technology during the last 20 years. This development has made a great impact in changing the procedures for composing and producing music within my genre without necessarily changing the strategies of how the technology is used. The transition from analogue to digital sound technology not only opened up new ways of manipulating and manoeuvring sound, it also opened up challenges in how to integrate and control the digital sound technology as a seamless part of my musical genre. By using techniques and methods known from electro-acoustic/computer music, and adapting them for use within my tradition, this research aims to find new strategies for composing and producing music within my genre.

Keywords

Artistic research, strategies for composition and production, convolution, environmental sounds, real time control

1. INTRODUCTION

The relationship between electro-acoustic and rock/metal music (as a part of the popular music umbrella) has a complex history relating to musical directions, intentions, the use of synthesis and manipulation. Nevertheless it can be said that both genres have embraced and integrated the technological tools made available at their present time. Even though there are several arguments pointing towards a blending of the use of technology between the genres, there are still many transfer possibilities and potential for exchange.

In my field the utilization of digital sound technology to a large degree still follows the same mindset that has been developed through the history of analogue sound technology.
It is therefore a large resource of unrevealed potential in contemporary technology for use within my genre.
In this research I address the following question:
How is it possible to transfer methods and techniques from one tradition to another without loosing the idiomatic characteristics of a genre, and how can you use this knowledge to add new aesthetics?

2. METHODS AND TECHNIQUES

Throughout the project the course of events has consisted of three main stages:
1. Studying and interpreting a selection of methods and techniques within electro-acoustic music and how to receive experience based knowledge of its possibilities and limitations.
2. Translating and adapting these methods and techniques for practical use within my genres aesthetics.
3. Proposing different ways of controlling these as an extended part of the instrumentation within my genre.

The focus areas throughout these different stages have been:
2.1 The studio as a compositional tool
2.2 Musical integration of environmental sounds
2.3 Creative use of convolution
2.4 Real – time control

2.1 The studio as a compositional tool

In maintaining the content of this progress, it was an obvious consequence to start with the sound studio as a framework and basis for several reasons. The sound studio has been a mutual point of focus and also a necessity for developing the aesthetics of both electro-acoustic and popular music. At the same time this meeting point divides these genres when it comes to working procedures. While electro-acoustic music has an acousmatic tradition being composed in a studio environment, the tradition within rock music is that recording normally takes place at the end of a composition process. In other words, the composing and rehearsal takes place in a dialog process between the different performers in real time. The use of the sound studio early in this process therefore leads to a challenge in how to maintain this dialog principle. In the electro acoustical tradition the division between the composer and producer has in some degree been absent. Within popular music this situation has been the opposite. In this case the producer becomes an important part of the 'so-called' music industry, and is given credibility as a part of the creative process. As early as 1978 Brian Eno talked about the obliteration of the composer/producer role within popular music when developing his ambient music.[5] He suggested that the sound studio as a compositional tool was one of the clearest characteristics in new music, and that this would become the main focus for compositional attention in the future.[3] Even though the DAW to a large degree has replaced the traditional recording studio, and the compositional procedures within popular music shifts against a use of the DAW earlier in the process, several of the mentioned conventions are still present. So how is it possible to reveal more of the potential in contemporary technology for use within my genre?

2.2 Musical integration of environmental sounds

The use of environmental sounds as building blocks in compositional works has been a significant progression within the electro acoustic tradition. Ever since musique concrete in the 50`s and up until today this direction has been developing, and is still a basis for different musical directions and expressions. This aesthetical approach is relatively unexplored within my genre. Working with environmental sound challenges the sonorous attributes within my genres conventional expression, but it also raises the question of how to control and integrate pre-recorded material as a part of a real time performance. In my research the selection of sounds has mainly been focused on industrial noise.[11]

Sound example 1:

This is a preview of a composition build up of drums, vocals, angle grinders, trains, boats and chains. The recordings of the angle grinders and trains are edited, tuned and organized as tonal instruments, the boats and chains as percussive instruments.

2.3 Creative use of convolution

Convolution tools have been available for composers since the early nineties[10]. In popular music they are most commonly used in reverberation units where they are based on recorded impulse responses from different rooms. These impulses are then stored in order to be convolved with a desired input. In addition to this approach there are no limits as to which sounds that can be convolved with each other, and the exploitation of these possibilities is where the research of this projects aims. Other examples of approaches to this technique is Roberto Aimi`s percussion instrument [1], or "the sound of touch" [4]. A more creative use of this technique can be found in some of Barry Truax`s works[13] within art music, or The Soundbyte`s "City of Glass"[12] within a popular music genre. As far as I am aware there has been very limited documented artistic research on convolving different sound sources with each other, and because of that most descriptions of use are focused more on technical than aesthetic aspects.[10] By using a wide variety of different environmental sounds as impulse responses this project has explored which possibilities and limitations convolution between digital sound files imply both at a micro level, but also in a broader musical context. This work has resulted in three different approaches.

2.3.1 Convolution in postproduction

The first approach to this work started with empirical experiments with a wide variety of pre recorded environmental sounds consisting of different attributes. The central aim through this experience was to be able to predict how different inputs and impulses would interact with each other, in order to control these parameters against a wanted output.

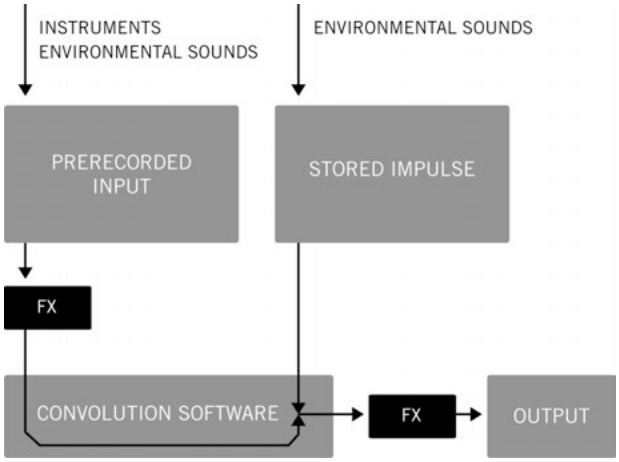


Figure 1. First setup

Sound example 2:

This is an example of a tonal approach, convoluting an electric guitar with the sound of a train**.**

Sound example 3:

This is an example of a rhythmical approach, convoluting handclaps with a recording of a chain

Sound example 4:

This is an example of both rhythmical and tonal approach put in a musical context

2.3.2 Real-time convolution

The second approach was finding ways to interact with this technique in real time by opening up a two-way communication between a musician and the output. In order to realize this two-way communication it was crucial that the musician was separated from the acoustic sound of the instrument in order to interact with the processed signal. This was maintained by feeding back the processed signal through headphones. By changing the impulse responses, and tailoring them to suit the present instrument, it was possible to affect the performance without the musician feeling unfamiliar with the mechanical presence or playing techniques of his own instrument. During these experiments both dry input signal and processed signal were recorded in order to analyze what caused the sonic changes, but also what made the musician make different artistic choices when interacting through this two-way communication.

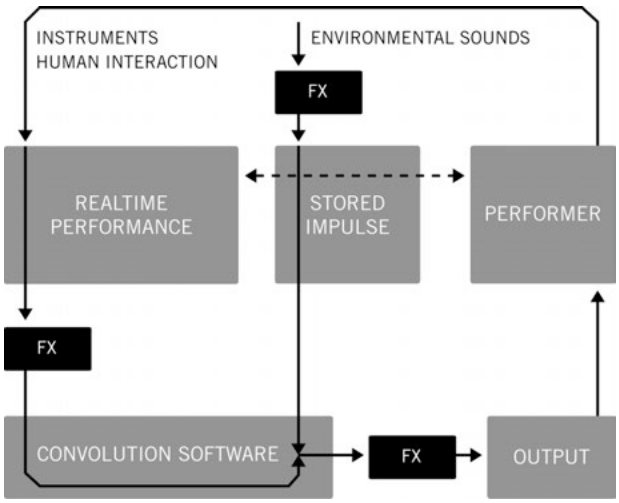


Figure 2. Second setup

Video example 1:
A video example of real time use of this set up (guitar and environmental sounds) together with real time convolution between drums and environmental sounds propose an artistic use of the points mentioned above.

2.3.3 The impulse sampler

The third approach came as a result of the experiences gained from the first two setups. The idea was to be able to record an impulse response and interact with it in a real-time situation. This setup gave the opportunity to sample impulses from my own instrument, other musicians or sound sources, and directly convolute them with another chosen sound-source in real-time.
The use of this setup gave several advantages. Firstly the implementation of this function made the whole process of trying different sounds against each other much faster and effective. Secondly the artistic value of being able to control samples of fellow musicians with my own instrument in real-time, opened up some exciting possibilities and results. The program was implemented in Csound, and runs in Ableton Live as a Max For Live device. [2]

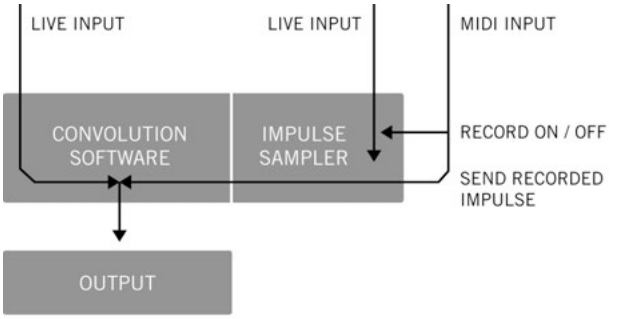


Figure 3. Impulse sampler
Sound example 5:
This is an example using the Impulse sampler to convolute a guitar with itself while playing.
Sound example 6:
This is an example using the Impulse sampler to convolute a guitar with an anglegrinder.

2.4 Real-time control

A challenge throughout the project has been finding ways to control these techniques in real time, and being able to use this in a musical dialog together with other musicians as an extended part of the conventional instrumentation. Since working in the studio in recent years to a large degree has changed from manoeuvring large mixing consoles to controlling everything through the DAW with a mouse and a keyboard, it felt natural to follow up on this workflow also in a real time situation. Even though there are several custom made interfaces for these operations on the market, few of them are made for integration on an existing instrument. As a guitarist both hands and feet are occupied at the same time concentrating on the guitar and foot pedals, disabling the player to handle a different standalone interface at the same time. The first step was to place a numerical keyboard directly on the guitar in order to control the DAW without interfering with the conventional playing. This solution opened up two different directions.

2.4.1 Controlling the DAW from the guitar

In a conventional guitar set up the closest solution for controlling a DAW lies in the use of a midi floorboard. Many of these floorboards already contain most of the functions needed for controlling both static and dynamic parameters in a software environment through its different stomp and expression pedals. At the same time this approach leads to a practical challenge in operating both the DAW and external hardware guitar processors at the same time from the same interface. The first approach was to attach a keypad directly onto the guitar in order to take care of the non-guitar operations in the DAW, and at the same time separate the control of the guitar processors and the DAW by using two midi floorboards.

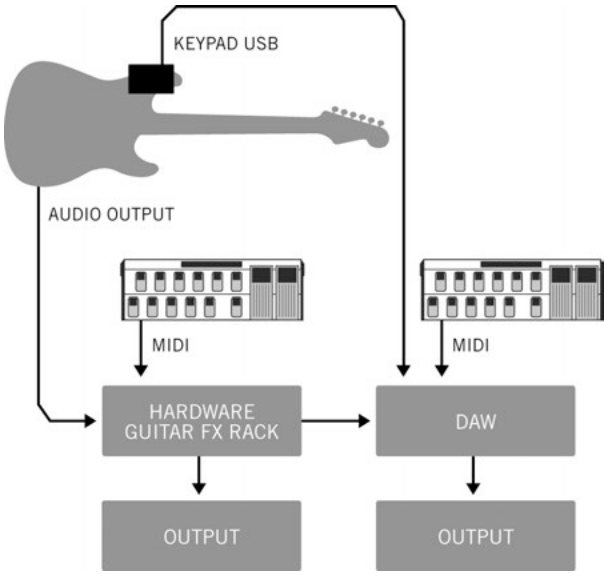


Figure 4. First setup

This figure can be seen as a miniature set up of a conventional studio event, and as a first attempt at bringing the traditional studio environment into the real time domain. Through this solution the traditional roles of the producer and musician are moulded together, but the system setup still consists of two parallel lines of control. This led to a search for a new solution where these roles were more seamlessly integrated with each other, and at the same time more individually flexible and comprehensive.

2.4.2 Augmentation of the guitar based on extended techniques

The functionality and practical use of contemporary digital guitar controllers are mainly based on a heritage stemming from electrical reproduction conventions, (different stomp boxes and expression pedals), resulting in a large amount of different digital floorboard and multi effects solutions. There are other approaches for digital augmentations of the electric guitar like the multimodal guitar[8] [9] or the Manson guitar[7]. Besides these there has been a limited documented research on digital augmentation solutions attached and controlled directly on the Electric guitar. At the same time the possibilities and functionality of tailor-made guitar software are poor compared to tools you find in most DAW programs, and it would therefore be natural to start with the DAW as a processing engine controlled from the guitar. From a musicians point of view it would be natural to integrate interfaces directly into the instrument, enabling real time control over the digital functionality without interfering with the playing of the instrument. The approach in this project has been to put together well-known and intuitive interfaces and to attach them directly to the instrument in order to control the digital software in real time. The direct integration of a keypad and a track pad enables a player to send both static and dynamical control

messages to different software and hardware in real time without removing the physical focus from the instrument or interfering with the idiomatic characteristics of the guitar. These interfaces are also very intuitive because of their use in other application on daily basis, and also quite inexpensive compared to custom-made solutions.

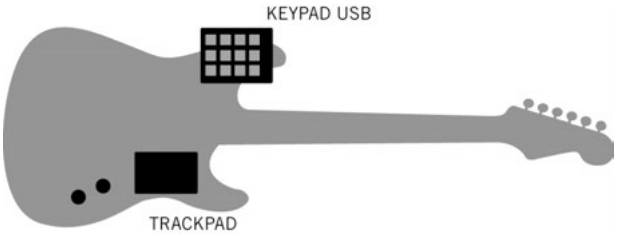


Figure 5. Placement of the interfaces

The physical placement of the two different pads was decided upon based on two well established extended, guitar techniques. The keypad was positioned in a typical guitar channel selector area, based on an on/off technique known as kill-switch.[6] The track-pad was placed in the volume/tone control area on the guitar based on an extended technique called volume swell. [14] The volume swell technique enables the player to use the volume knob dynamically without removing the right hand position from the instrument. This was the basis for the second guitar setup.

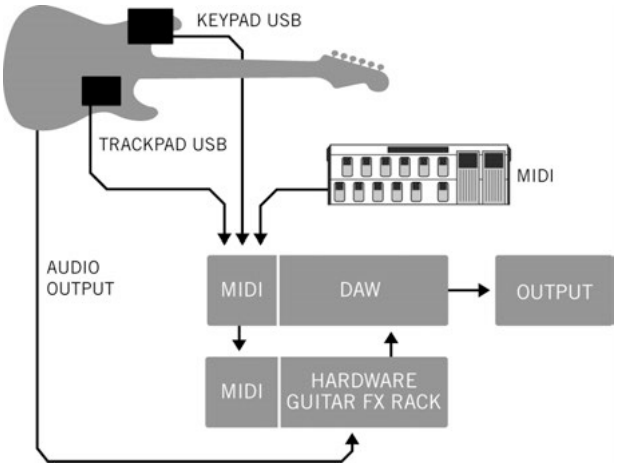


Figure 6. Second setup

This set up gave several advantages. First of all it gave the possibility of removing some of the components from the first setup without compromising the DAW control or preset changing in the Guitar-FX hardware. This was done by running all incoming control messages from the different interfaces directly through the DAW for mapping and further distribution. Secondly, the track-pad opened up an easier and more intuitive way of controlling XY parameters compared to using two expression pedals at the same time. The physical placement of the track-pad also contributed to the possibility of using the XY parameters without removing the right hand position as in contradiction to the Manson guitar system.[7]

2.4.3 Interface output

Both the keypad and track-pad outputs are translated to midi signals through two different Max For Live devices[15]. The keypad can be used to perform static operations like on/off and momentary messages. The track-pad can be used to perform dynamic operations like volume, morphing between different effects, surround sound operations or other applications demanding XY control.

Video example 2:
Demonstration of the guitar setup, using the track-pad as an XY controller in a granular synthesis plug in.

Sound example 7:
Demonstration of the guitar setup in a real-time improvisation with other musicians playing convoluted piano and percussion.

3. SUMMARY

This is still an ongoing research project, where all mentioned themes and work are constantly under a refinement process. The next step is to proceed with the research through an even more practical approach. This will be done by recording and doing concerts with different musicians within a real time context for experiencing points for further technical and aesthetical improvements.

4. REFERENCES

[1] Aimi R. M.(2007) ”Hybrid Percussion : Extending Physical Instruments Using Sampled Acoustics” PhD thesis, Massachusetts Institute of Technology.

[2] Brandtsegg, Øyvind (2011): ”The Impulse sampler was implemented in Csound and Max For Live by Øyvind Brandtsegg” oyvind.brandtsegg@ntnu.no

[3] Cox, Christoph and Warner, Daniel (2004): “Audio Culture: readings in Modern Music”, The continuum International Publishing group Inc.

[4] D. Merrill, H. Raffle, R. Aimi. (2008) “The Sound of Touch: Physical Manipulation of Digital Sound”. In the Proceedings the SIGCHI conference on Human factors in computing systems (CHI'08). Florence, Italy.

[5] Eno, Brian (1978):“Interview with Brian Eno”, viewed 10. April 2011, http://music.hyperreal.org/artists/brian_eno/interviews

[6] Killswitch, viewed 10. April 2011, http://www.instructables.com/id/Guitar-Killswitch-Strat.-design/

[7] Manson guitar, viewed 10. April 2011, http://www.mansonguitars.co.uk/

[8] Multimodal guitar, viewed 10. April 2011, http://www.numediart.org/projects/07-1-multimodal-guitar/

[9] O. Lahdeoja (2008). An approach to instrument augmentation : the electric guitar. In Proc. of the 2008 Conf. on New Interfaces for Musical Expression (NIME08).

[10] Roads, Curtis (1996): “the computer music tutorial”, The MIT Press.

[11] Russolo, Luigi (1913): “The art of noises”

[12] The Soundbyte/Irgens (2007), City of Glass, Voices Music Publishing

[13] Truax, Barry, viewed 10. April 2011, http://www.sfu.ca/~truax/conv.html

[14] Volume swell, viewed 10. April 2011, http://en.wikipedia.org/wiki/Volume_swell

[15] Wærstad, Bernt Isak (2010): “The trackpad translator was implemented in Max For Live by Bernt Isak Wærstad” http://partikkelaudio.com/extras/mfl/

5. Appendix

All sound and video examples can be found at:
http://thesoundbyte.com/nime

The *Six Fantasies Machine – an instrument modelling phrases from Paul Lansky's* Six Fantasies

Andreas Bergsland
Dept. of music, NTNU
7491 Trondheim
Norway
andreas.bergsland@ntnu.no

ABSTRACT

The *Six Fantasies Machine* (SFM) is a software instrument that simulates sounds from Paul Lansky's classic computer music piece from 1979, *Six Fantasies on a Poem by Thomas Campion*. The paper describes the design of the instrument and its user interface and how it can be used in a methodological approach called *the epistemology of simulations* by Godøy. In *imitating* phrases from Lansky's piece and enabling the creation of *variants* of these phrases, the user can get an experience of the essential traits of the phrases. Moreover, the instrument will give the user hands-on experience with processing techniques that the composer applied, albeit with a user-friendly interface.

Keywords

LPC, software instrument, analysis, modeling, csound

1. INTRODUCTION

The Six Fantasies Machine (SFM) was developed as a part of my doctoral project, *Experiencing Voices in Electroacoustic Music* [1]. Here, I establish a framework for understanding and describing the listener's experience of voices in acousmatic electroacoustic music and related genres. The framework is then applied in evaluating and describing Paul Lansky's classic computer music work from 1979, *Six Fantasies on a Poem by Thomas Campion.* In addition to using this framework, which is largely based on a phenomenological and introspective methodology, I also explore how variations of the musical phrases appearing in the piece affect the listening experience.

The SFM was developed to imitate or simulate the phrases appearing in the piece, but also to be able to synthesize variants closer or further from the original phrases, and to explore how such variants can affect the experience. This paper will briefly discuss the theoretical context of my overall methodology. Furthermore, I will describe relevant features of Lansky's work and its composition process, before giving a more detailed explication of the technical layout and user interface of my instrument. Subsequently, I will show how the instrument has come to use in my project, and finally point to other uses of the instruments and possible ways to develop it further.[1]

[1] The SFM instrument and the thesis are available for download at http://folk.ntnu.no/andbe/Projects.html

2. EPISTEMOLGY OF SIMULATIONS

In his dissertation Godøy has developed the notion of the *epistemology of simulations* [2]. By this, he refers to the possibility of creating variants of a sound event or object where a certain trait or aspect is changed while others remain the same, and then *observing* and *comparing* the effect that this has on the experience. This is seen as a means of seeing what constitute the essential traits of musical objects. The epistemology of simulations is usually at work in the paradigm of synthesis, Godøy argues, because synthesis usually involves interplay between *listening* to the sound and, by trial and error, *tuning* the synthesis model and its parameters so as to achieve the desired result. However, Godøy also suggests that this approach could be applicable in the investigation of musical objects on a larger scale, such as phrases in a musical work. There have been some examples of studies that use a methodology not too far from what Godøy suggests.

Firstly, Michael Clarke has developed an interactive software tool for analyzing Jonathan Harvey's computer music work *Mortuos Plango, Vivos Voco* [3]. The software is meant to aid listeners in their exploration of the composition by allowing them to recall specific locations from the work and to juxtapose these, but more interesting in this context is that it also allows for interactive exploration of the synthesis and processing techniques used in the piece. Clarke argues that since such techniques are often a part of the individual work, it is important for someone wanting a deeper understanding of the music to clarify the components of the compositional technique. Trying out the techniques in an interactive manner might learn the user more about the potential of the techniques so as to place the compositional choices made by the composer in a wider context of possibilities [4].

Secondly, Keller and Ferneyhough develops what they call *analysis by modeling*, in their study of the temporal quantization processes and the streaming-fusion processes in Xenakis's *ST/10-1 080262*, one of the first computer-generated algorithmic compositions [5]. The stochastic algorithms of this piece is implemented by the authors in *Patchwork* (IRCAM), thus opening for a number of different realizations of the piece. This is then used to investigate temporal quantization and streaming-fusion processes at work in perception.

In that way, both approaches have many similarities with Godøy's notion of an *epistemology of simulations*, albeit in somewhat different ways. While Clarke's approach stresses the pedagogical and hermeneutical goals in developing a tool for simulating musical objects, Keller and Ferneyhough focus more on particular aspects of perception and how these are affected by manipulating the parameters of the model. I will show in section 5 that the SFM instrument has components from both these approaches.

3. LANSKY'S *SIX FANTASIES*

3.1 Compositional idea and layout

As the full title suggests, *Six Fantasies* is based on an untitled poem by the English Renaissance poet Thomas Campion (1567-1620), published in his treatise *Observations in the Art of English Poesie* in 1602 [6]. In accordance with many so-called text-sound pieces, the recitation of the poem is recorded, and with few exceptions the recorded recitation constitutes the sound source material for the whole piece [7]. In the five first movements of the composition, entitled *her voice*, *her presence*, *her reflection*, *her song* and *her ritual*, the recitation is manipulated and temporally modified so as to create different sonic manifestations of the poem. The original reading is then presented in the last movement, *her self*, accompanied only by synthetic sounding stretched out vowels.

3.2 Technical procedure

The main technique that Lansky applied in this piece was Linear Predictive Coding (LPC), developed during the 1960s and 70s, mainly to compress speech signals [8, 9]. Greatly simplified, the LPC *analysis* of the speech signal makes an estimation of the time-varying filter component, the fundamental frequency of the phonation component, the intensity of the signal, and whether the signal is voiced or un-voiced, i.e. noisy [10]. In the *resynthesis* process, the error signal from the analysis is used to decide whether to synthesize a buzz signal (pulse-train) for the voiced parts, or white noise for the un-voiced part. The appropriate signal is then controlled by the analysis parameters for intensity (buzz and noise) and fundamental frequency (buzz only). The resulting source signal is then fed through a time-varying filter controlled by the analysis parameters, so as to create a synthesized approximation of the speech signal. By changing the analysis parameters, it is possible to manipulate the source component, in particular fundamental frequency (f0) and intensity, independently of the filter component. Since the analysis is made on a frame-by-frame basis, it is also possible to vary the frame rate in the re-synthesis process, thereby changing the playback speed without affecting pitch or spectrum. This enabled Lansky to transform the recitation in different ways and to different degrees; transposing, inverting, flattening, exaggerating or fully "sculpting" the intonation contour, time-stretching the signal and replacing the buzz with noise.[2] The reading/speaking voice could be thereby be transformed into what sounded like a kind of singing (as in *her song*), a vocal style between speech and song (as in *her presence*), and whispering (as in *her ritual*). Lansky also implemented a "chorus" effect by using several pulse generators together with small random variations in fundamental frequency, hence creating a richer sound [Lansky, personal communication].

In addition to the LPC technique, Lansky also applied comb filters extensively in *her reflection* and *her ritual*. Each comb filter, which is commonly implemented as a delay with feedback added to the original signal, adds a resonance at a particular frequency depending on the delay time and amount of feedback. By using banks of many double comb filters, Lansky could produce rich resonances, often with a chord-like flavour. By setting the delay time higher, however, Lansky could create more typical delay effects, as in *her reflection*. A small amount of reverberation can also be heard in several of the fantasies.

[2] The possibility of shifting the spectral envelope (i.e. the filter component) that LPC offered, however, was only applied in the accompaniment of *her self*.

4. THE SFM INSTRUMENT

The SFM instrument is developed in the script based synthesis and processing environment *csound* [11]. This environment was used both to implement the signal processing parts as well as the graphical interface (GUI). It has been developed and tested for Windows XP, but should in principle be possible to run on other platforms with minor adjustments.

Although the SFM instrument produces vocal phrases that are similar to those that can be heard in Lansky's piece, they are *not* based on the same vocal material. Having no access to Lansky's original LPC files nor to his sound files, I have instead used an actress to imitate the original reading as it is presented in *her self*. The recording of her reading was then resampled to 14 kHz, the same as Lansky originally applied, analyzed with the LPC analysis utility in *csound*, LPANAL, and used as a basis for the LPC-resynthesis. Lastly, SFM is in its current version equipped with 8 voices, which can start simultaneously or with a delay of up to 10 seconds.

4.1 Resynthesis and processing

SFM uses the same standard implementation of the LPC resynthesis as in *Six Fantasies*. A conditional statement assessing the error value from the appropriate analysis file read by the `lpread` opcode decides whether a voiced (`buzz`) or an unvoiced noise source (`rand`) is passed through a time-varying filter (`lpfreson`).[3] A scaling value that crossfades between the buzz and the noise signal also enables blending the buzz and noise signals, or sending only noise through the filter.

The time-pointer, with which the analysis file is read, is divided into six segments, and the break-points defining these segments are editable by the user. By multiplying the duration value of each these segments with a factor, independent time-stretching/compression of segments can be achieved.

While the analysis file provides pitch and amplitude analysis of the signal which can be used to control f0 of the `buzz`-opcode, the instrument also gives possibilities of either manipulating the values from the analysis or setting them independently. A scaling value weighs the values from the analysis file against those provided by the user, thus enabling an intermediate situation between these two. The f0 values can be multiplied with a factor and thereby transposed. Moreover, by scaling the deviation around a calculated mean f0, the f0 contours can be flattened, exaggerated and/or inverted. The f0 values can also be set independently for each of the segments, and by applying a low-pass filter the length of the glide between the static f0 values can be adjusted. A mechanism for tuning the user provided f0 values into the tempered scale is also implemented in the instrument.

The `lpfreson-`opcode opens for shifting the frequencies of the filter up or down, so as to transpose the spectral envelope and the vocal formants of the resynthesized sound. A small amount of band pass filtered noise is also added to the buzz source to prevent a "buzzy" quality and thereby increase naturalness. Moreover, "chorus" is implemented as an optional feature, mixing the original signal with the output of four additional buzz sources with small random pitch variations (+/- 0.8%).

The resynthesized signal is then fed through an instrument implementing a bank of 9 double comb filters per voice, making it possible to use up to 72 comb filters simultaneously. Finally, the signal passes through a reverberation instrument before it is sent to the output.

[3] A simplified flow chart of the instrument is available at: http://folk.ntnu.no/andbe/Projects.html#SFM-Flowchart

4.2 GUI

The user interface is implemented with the Fast Light Toolkit (FLTK) opcodes in *csound*. A snapshot of the whole interface is shown in Figure 2. The interface is split into two sections, the *voice section* and below it, the *mixing section*. The *voice section* contains the controls for the eight individual voices, ordered as tabs. At the bottom left of the section, the user can choose between the 21 available LPC analysis files, which contain the vocal phrases of the poem. Choosing a file will at the same time show the appropriate text and the duration of the file. At the top left of the section, the user can set the seven break points that define the six segments of the analysis file.

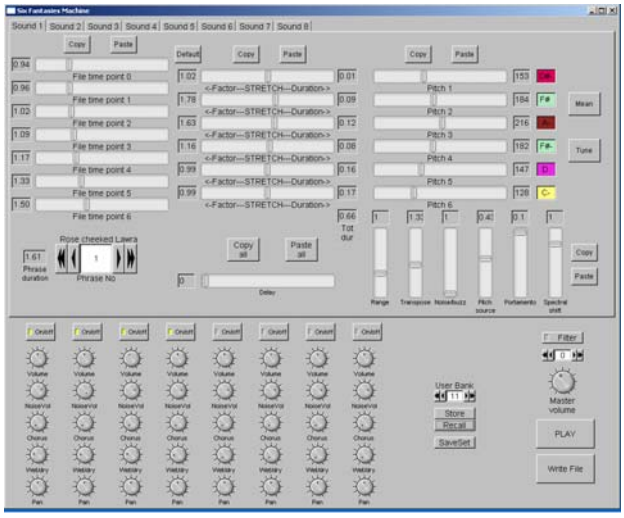


Figure 2. SFM GUI with *voice section* at the top and *mixing section* at the bottom.

These segments can then be time-stretched or compressed using the stretch factor controllers in the middle. The stretch factor values as well as the duration of the segment will show up in number boxes next to the sliders, and the latter will be summed to display the total duration of the phrase.

At top right of the section, pitch/frequency for each of the six sections can be set (see figure 3 for details). Here, the pitches closest to the frequency values chosen will turn up in a colored window next to the slider, with different colors for each pitch. "+" and "-" indicate whether the pitches are slightly high or low. Pressing the "Tune" button will then set all the chosen frequencies to correct tempered pitches, thus removing any "+" or "-" signs. The "mean" button will set the sliders to the mean f0 of the analysis file. The lower right part of the *voice section* contains sliders for setting f0 *range* (scaling of deviation around mean f0), *transposition*, *noise/buzz mix* of the source signal, *portamento* (glide) time between consequent f0 values set by the user, and *spectral shift*. At bottom centre the time delay of the voice can be set. For easy transfer of values between voices, copy and paste buttons are provided for both slider groups and for all the voice controls.

The mixing section in the lower part of the interface is split in three parts. To the left, there are eight mixing "stripes" for each of the eight voices, with "on/off", volume, noise level, chorus, reverb and pan controls. To the right, the user can press a button to activate the comb filter instrument and choose between filter presets. The setting for the comb filters can in the current version be edited by opening accompanying text files in a spreadsheet editor. Below the master volume, there are

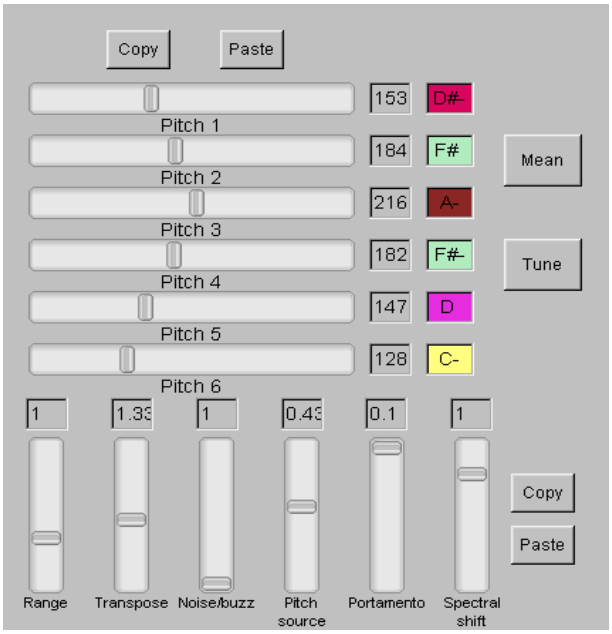


Figure 3. SFM GUI detail from *voice section*

buttons for playing back the phrase with the current setting and for writing this to a sound file. Finally, between these sections there are buttons for saving and loading presets.

5. USING THE SFM

Since the SFM was developed as a tool aiding an epistemology of simulations, creating imitations of phrases from Lansky's piece and varying these has been my main focus area. However, I have also found that the instrument can be used as a tool for creating sound material for new compositions. As in Clarke's project on *Mortuos Plango* all these ways of using the instrument can give its user a better understanding of the techniques used by Lansky when composing *Six Fantasies*. I will address these areas of use in turn.

5.1 Imitations

Creating vocal phrases that are relatively similar to phrases in *Six Fantasies* has been one important goal in the process of developing the instrument. Without sufficiently similar sounding results it would be very difficult to establish the link to Lansky's piece. In its current form, I experience that this link is indeed present, and this is confirmed by others who had heard the results, including Lansky himself. After having worked extensively with a selection of phrases that I wanted to imitate I have come to the following conclusions:

- Phrases from most movements can be imitated fairly well.
- Many of the phrases in *her ritual* have been difficult to imitate due to limited possibilities for repeated triggering.
- Factors as a) using another voice than in the original piece, and b) not knowing anything about the spectral characteristics of the recordings used to create Lansky's LPC-files, have delimited the degree of similarity that can be attained.
- Imitating the phrases has facilitated a fine tuning of perception, thus giving access to minute details of the piece.
- Making the SFM has given me an increased understanding of the technical challenges and sounding results of both LPC and comb filters as well as Lansky's compositional process.
- The lack of precise parameter control due to a limited number of sliders makes the imitations less similar than when controlling the parameters via a text-based score file. Still,

the lack of user-friendliness in editing parameters as text makes it an alternative only for the particularly interested.

In the current version, ten of the imitations that I have made are included as presets in the instruments.[4] New users will probably also learn something about Lansky's piece both by looking at the imitations included in the presets and by trying to create new imitations from scratch.

5.2 Variations

Starting from the imitations, one can create many types of variations of the phrases, which allow the user to learn how the different control parameters can affect the sound. In particular, it can be interesting to:

- decompose the phrases into individual voices
- add new voices, e.g. in new transpositions, with flattened, exaggerated or inverted pitch trajectories
- stretch or compress shorter segments or whole phrases
- change one or several pitches in phrases which are controlled from the pitch sliders
- modify the resonant frequencies or the decay times of the comb filters where these are applied

Playing with such variations and comparing them to the original phrases and/or their imitations may then make the user experience that there are limits to the degree of modifications that can be applied before the phrases turn into "something else". Thus, such experiences can give a sense of what makes the phrases what they are – in other words the "essential traits" that Godøy talks about (see section 2).

5.3 Exploring aspects of listening

Working with my PhD project, the epistemology of simulations has been one of several methodological strategies to help establish a framework for describing and understanding the experience of voices in electroacoustic music. The SFM instrument is one of several tools that have demonstrated in what ways different control parameters can affect different aspects of the experience, for instance: a) the directing of attention towards different aspects of the vocal phrases, such as verbal semantic, affective or identity related aspects, b) information density, c) naturalness/artificiality, d) salience, or e) stream integration/segregation.

5.4 Composing

The most enjoyable way of using the instrument is probably playing freely with parameters to create results that are sonically interesting, be they closer or farther from the phrases in *Six Fantasies*. These phrases can then be written to file and used as sound material for electroacoustic works of music.[5] Working with the instrument in this way will probably also expand the users' knowledge of, and a feeling for, the sonic results produced by the LPC resynthesis, the comb filters and the other parameters. Thus, it might still indirectly increase the users' understanding of the tools applied by Lansky in his original composition, even though the interface facilitates a much more accessible means of creation and modification.

6. FUTURE DEVELOPMENT

There are several features of the SFM instrument that have room for improvement in future versions:

- Increase the number of voices
- Implement widgets for controlling the comb filter parameters in the GUI.
- Increase the number of comb filters in the instrument.
- Increase the number of pre-made presets, as well as the number of presets available for editing by the user.

Currently, I am also considering creating a GUI for the instrument in *Max* by using the *csound~* external written by David Pyon [12,13]. This opens up for letting the user work with the pitch contours graphically through the *multislider* object, thus making it possible to see and edit the pitch contours with the same interface. Porting the GUI part of the instrument to *Max* also opens up for other interesting possibilities. By adding text, pictures and video one could develop an interactive multimedia pedagogical tool that approaches for Lansky's *Six Fantasies* what Clarke did for Harvey's *Mortuos Plango*.

7. CONCLUSION AND FUTURE WORK

The SFM instrument can in its current version be used in an *epistemology of simulations* approach, as Godøy delineated. Thus, the instrument can be a valuable tool in guiding a user towards an expanded understanding of Lansky's *Six Fantasies* through active exploration. In particular the instrument can give the user hands-on experience with signal processing techniques the composer used, albeit with a user-friendly interface. I have also shown how the instrument can also be used to illustrate experiential aspects of voices in electroacoustic music. Lastly, the SFM instrument can itself be applied as a compositional instrument for composers and sound designers interested in exploring the sound world of techniques which in today's musical world might seem obsolete and outdated, but which nevertheless provides a sonically rich palette of expressions.

8. REFERENCES

[1] Bergsland, A., *Experiencing Voices in Electroacoustic Music*, in *Dept. of Music*. 2010, NTNU: Trondheim.

[2] Godøy, R.I., *Formalization and Epistemology*. 1997, Oslo: Universitetsforlaget, 295-296.

[3] Clarke, M., *Jonathan Harvey's Mortuos Plango, Vivos Voco*, in *Analytical Methods of Electroacoustic Music*, M. Simoni, Editor. 2005, Routledge: New York. p. 111-143.

[4] Clarke, M., *An Interactive Aural Approach to the Analysis of Computer Music*, in *International Computer Music Conference*. 2005: Barcelona, Spain.

[5] Keller, D. and B. Ferneyhough, *Analysis by Modeling: Xenakis's ST/10-1 080262*. Journal of New Music Research, 2004. 33(2): p. 161-171.

[6] Campion, T., *Observations in the art of English poesie, 1602*, in *A defence of ryme against a pamphlet entituled Obseruations in the art of English poesie, 1603*, G.B. Harrison, Editor. 1966, Barnes & Noble: New York.

[7] Ondishko, D., *Six Fantasies on a Poem by Thomas Campion: Synthesis and Evolution of Paul Lansky's Music Compositions.* 1990, Eastman School of Music: Rochester, New York.

[8] Atal, B.S., *The History of Linear Prediction.* IEEE Signal Processing Magazine, 2006(March): p. 155-161.

[9] Atal, B.S. and S. Hanauer, *Speech Analysis and Synthesis by Linear Prediction of the Speech Wave.* The Journal of the Acoustical Society of America, 1971. 47(2B): p. 637-655.

[10] Lansky, P., *Compositional Applications of Linear Predictive Coding*, in *Current Directions in Computer Music Research*, M.V. Mathews and J.R. Pierce, Editors. 1989, MIT Press: Cambridge, Mass. p. 5-8.

[11] Boulanger, R., ed. *The Csound Book*. 2000, The MIT Press: Cambridge, Mass.

[12] http://cycling74.com/products/maxmspjitter/

[13] http://davixology.com/csound~.html

[4] Available online at http://folk.ntnu.no/andbe/Projects.html.

[5] One short "etude" made by this author is available at http://folk.ntnu.no/andbe/Projects.html.

Gliss: An Intuitive Sequencer for the iPhone and iPad

Jan Trützschler von Falkenstein
University of Birmingham
TeaTracks
The Hague, The Netherlands
jan@teatracks.com

ABSTRACT
Gliss is an application for iOS that lets the user sequence five separate instruments and play them back in various ways. Sequences can be created by drawing onto the screen while the sequencer is running. The playhead of the sequencer can be set to randomly deviate from the drawings or can be controlled via the accelerometer of the device. This makes Gliss a hybrid of a sequencer, an instrument and a generative music system.

Keywords
Gliss, iOS, iPhone, iPad, interface, UPIC, music, sequencer, accelerometer, drawing

1. INTRODUCTION
Gliss is an intuitive music sequencer, which lets the user create and perform music.
Sequences can be created by drawing onto the screen which can then be performed by tilting the device. The x-axis of the accelerometer controls the position and speed of the playhead, while the y-axis can be used to randomize an offset for each drawing. Thus drawings of sound can be performed in an intuitive physical way.

The idea of developing this app came while developing a prototype 16-step sequencer for the iPhone. Apart from the number of existing apps that implement quite sophisticated step sequencers, I was searching for a way to create loose and not quantized sequences, which could be interpreted in different ways. Using the accelerometer as unique feature of mobile devices we added the ability to scan back and forth through sounds on a time line and thus make it possible to approach a sequence literally from different angles.

2. BACKGROUND
The method of drawing sounds on a timeline has been explored by Iannis Xenakis with his UPIC (Unite Polyagogique Informatique du CeMaMu) system in the 1970s. The UPIC was a hardware device, which allowed the user to draw sound events, which were then synthesized by a computer. Since then a few other programs have been developed based on drawing sound. [4] [6]

Xenakis' UPIC system included an option to specify the amplitude over time. Such a feature has not yet been implemented in Gliss, mainly for the reason to keep it simple at the current stage of development.

Other approaches of using drawing for music creation include the installation and iPhone app Sonic Wire Sculptor by Amit Pitaru and DrawSound by Kazuhiro Jo. [2] [3] In contrast to Gliss these two systems not based on a straight timeline. Although Sonic Wire Sculptor does implement some kine of three dimensional timeline, it will produce sound while the drawing is still being made. DrawSound is not based on a timeline at all, but translates drawings immediately.

Gliss provides a facility for composing a piece with different section and performing it, which turns the iPhone into a musical instrument. Using the iPhone as an expressive musical instrument has been inspired by the work of MoPhO (Mobile Phone Orchestra). [5]

The option to randomize the x-position of the drawings by a tilt injects ideas of generative and algorithmic music into Gliss. [1] Though there is currently no method to let the Gliss app generate new events automatically, the simple mechanism of physically controlling a random frequency offset can turn it into a generative system.

In comparison to other generative music apps, Gliss tries to give the user more control over the musical outcome while still keeping a certain playfulness. By providing a general framework to create composition which can be performed by others it can be seen as a step towards mobile music as an emerging form of creating or listening to music which asks for physical action and will never sound exactly the same.

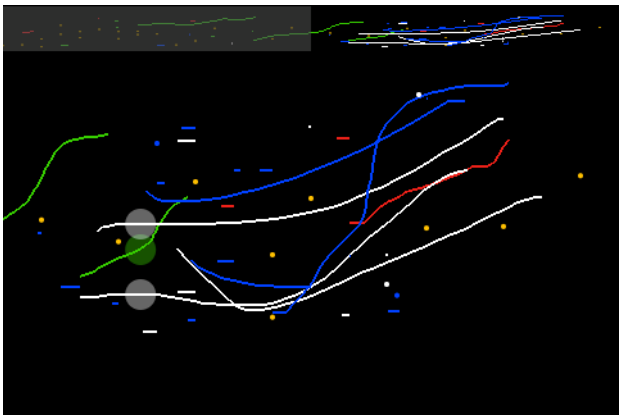


Figure 1. A screenshot of Gliss' main screen. Each color represents a different sound.

3. DRAWING ON A TIMELINE
Gliss has one main screen, a timeline, onto which sounds can be drawn in five different colors. Currently one can choose from three different instruments: a sound file player with variable playback rate, a sine oscillator and a sampler which assigns the y-axis of the screen to different samples rather than

to pitch. Samples and sound files can be loaded into the application via wifi or the internet (using SoundCloud, a commercial service to host and share music files). Contrary to most other sequencers currently available for mobile devices, Gliss is not based on a time grid. The placement of sound objects does not have to be quantized, sounds can be placed freely in time. However, there is an option to use a grid if a stricter rhythmical sequence is desired. This set-up allows a far more experimental approach than those offered by other sequencers, which are mostly designed for the creation of beats and dance music.

All sounds are placed overlapping on one screen, which means that there aren't any separate tracks for each color or instrument. Still they can have different ranges of pitch assigned to their y-axis. This has the advantage that one can easily have an overview of the sequence on small screens. Gliss works with project files, each of which can contain an indefinite amount of sequences. Each sequence can have its own samples, instruments and other settings, such as tempo. A piece can be performed by paging through the sequences.

4. PERFORMING SEQUENCES

Presently the app acts like a music sequencer. The special feature is that the playhead on the timeline and the position of the objects can be controlled by the movements of the hands.
Currently there are two modes to control the playhead: Via a fixed tempo, so that tilting the device changes the playback direction, or a physical behavior can be assigned, so that not only the direction but also the speed of the playhead changes according to simple, yet effective (virtual) mechanics.

A small button on the main screen lets the user switch between both modes. When pressing the button in 'physical' mode one can either freeze a certain speed in order to preserve the current tempo, or with a double tap return to a predefined tempo.

Another way of interacting with the composition is to randomize the y-position of the drawings. This feature can be toggled by a button in the main screen and controlled by tilting the device. Depending on the degree and direction of the tilt, the drawings after being played move to a new position within a random range. Holding the device flat sets a full range, so that the drawings can move anywhere on the y-axis of the screen. By changing the angle one changes also the range, resulting in objects more likely dropping towards the bottom. This creates randomness controlled by gravity. Similar to the tempo button, the current state can be frozen with a single tap or brought back to the original with a double tap.

5. CONCLUSION

Primarily the performing features in Gliss turn it into an instrument with which fixed composition can be performed and interpreted on the fly. Thus, when creating a sequence, one does not write or draw a fixed composition to be played back, but one rather composes some kind of prototype for a piece, which offers various variations one can listen to. Such method or style of music creation and reception could be described as Mobile Music, variable fixed composition which needs to be played on a mobile device. An important aspect hereby is that production, performing and listening happens on the same environment: the device and software.

Such a playful approach also attracts children who are often very open for experiments and shows them another angle of dealing with sound and defining music.

Gliss was also used in ensemble settings, where even untrained performers were able to play in a group.

For the more traditional producer it can serve as a sample and effects manipulation unit.

6. ACKNOWLEDGMENTS

The graphic design for Gliss and TeaTracks is done by Gabriele Hultsch and Studio Copernicus. Many thanks to Julio d´Escrivan, Scott Wilson, Nick Collins, Fredrik Olofsson for testing and suggestions and to all other beta testers.

7. REFERENCES

[1] N. Collins, The Analysis of Generative Music Programs, In *Organised Sound 13*. Cambridge 2009.

[2] Kazuhiro Jo. DrawSound: A Drawing Instrument for Sound Performance, In *Proceedings of the Second International Conference for Tangible and Embedded Interaction*. Bonn, 2008.

[3] A. Pitaru. *Sonic Wire Sculptor*, http://sws.cc/.

[4] Thiebaut, Healey, Kinns. Drawing Electroacoustic Music, In *Proceeding of the International Computer Music Conference*. Belfast, August 2008.

[5] G. Wang, G. Essl and H. Pentinnen. MoPhO: Do Mobile Phones Dream of Electric Orchestras?, In *Proceedings of the International Computer Music Conference*. Belfast, August 2008.

[6] Xenakis, I. *Formalized Music*. Stuyvesant, NY: Pendragon Press. 1992.

Quadrofeelia – A New Instrument for Sliding into Notes

Jiffer Harriman
CCRMA
Stanford University
jiffer8@ccrma.stanford.edu

Locky Casey
CCRMA
Stanford University
lauchlan@stanford.edu

Linden Melvin
CCRMA
Stanford University
lmelvin@stanford.edu

ABSTRACT
This paper describes a new musical instrument inspired by the pedal-steel guitar, along with its motivations and other considerations. Creating a multi-dimensional, expressive instrument was the primary driving force. For these criteria the pedal steel guitar proved an apt model as it allows control over several instrument parameters simultaneously and continuously. The parameters we wanted control over were volume, timbre, release time and pitch.
The Quadrofeelia is played with two hands on a horizontal surface. Single notes and melodies are easily played as well as chordal accompaniment with a variety of timbres and release times enabling a range of legato and staccato notes in an intuitive manner with a new yet familiar interface.

Keywords
NIME, pedal-steel, electronic, slide, demonstration, membrane, continuous, ribbon, instrument, polyphony, lead

1. Introduction
For an instrument to be expressive, it requires several degrees of freedom for the performer. The ability to control volume, timbre and pitch in a continuous way is paramount. The goal of this project was to create an electronic instrument that matched the expressivity of the pedal-steel guitar. The pedal steel has the ability to bend individual notes of a chord while keeping the rest stable to form new chords. This method, not readily available on current electronic instruments, is featured on the Quadrofeelia.

2. RELATED WORK
The GXTar [3] interface uses a similar membrane sensor to capture continuous data from the left hand. However Quadrofeelia uses a different method for activating notes and has a second set of controllers which affect pitch. Additionally, it is oriented horizontally on a table top instead of being held like a guitar.
Many solutions have been offered that provide means of bending notes, including the pitch bend wheel and various ribbon controllers such as the Kurzweil RBN1 [2]. What isn't readily available is that which makes the pedal-steel guitar unique, the ability to bend individual notes of a chord, thus changing the chord type or voicing, as opposed to moving all the notes in unison. A previous offering geared towards keyboardists is "The Glide" [1].

3. DESIGN OVERVIEW
3.1 Modes of Expression
To achieve a comparable level of versatility in an electronic instrument continuous controllers are needed. The choice of membrane position sensors was ideal because of their continuous output range and intuitive interaction. By adding an additional force sensitive resistor (FSR) strip under the position sensor, both pressure and position can be measured simultaneously. By using position sensors for both hands to control multiple notes chords can be formed and altered in a continuous way. Lastly, additional FSRs are placed below where the palm of the hand rests to provide means to mute ringing notes to varying degrees.

2.2 Technology
The instrument uses the BeagleBoard development board and the Arduino Nano platform for I/O. The BeagleBoard is running an Ubuntu Linux distribution which enables connecting to and executing programs through an SSH connection. Installed on the Beagleboard is Pure Data (PD) which is used for interpreting the sensor data from the Arduino as well as sound synthesis. The use of the BeagleBoard for sound synthesis creates a more portable self-contained instrument. The current iteration has increased latency than with the same patch running on a more powerful computer.

4. Interaction
4.1 Control
The controls for the instrument are one long (500mm) horizontal and four short (100mm) vertical strips and a panel of four buttons. Since the instrument is modeled after a string instrument, the left hand slider is marked with vertical lines indicating musical half-steps, and dots as is typical with guitar fretboards. While the left hand has completely continuous behavior the right hand interface is partitioned off into 4

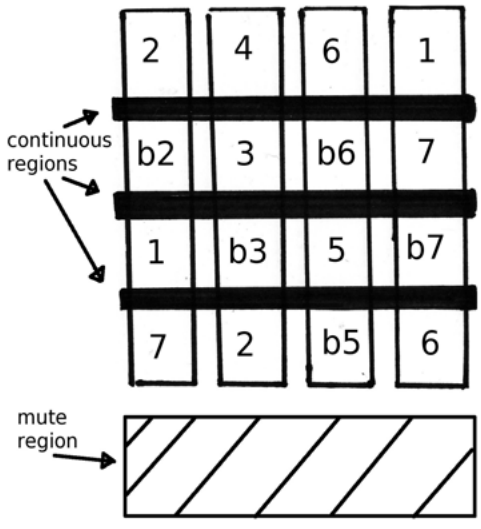


Figure 1: Right Hand interface control
Numbers indicate scale degree for initial tuning

sections per slider. Within a defined region the note does not change which makes it more forgiving. Between the static regions is a continuous region where the note will gradually change to the new region. This combination provides

NIME'11, 30 May–1 June 2011, Oslo, Norway.

flexibility to be able to change notes gradually or stay locked into a set of notes.

4.2 Right Hand

Four buttons select between different key mappings, or "tunings". The mappings for the right hand were chosen to provide a simple means of playing diatonic chords. The mapping associated with the first button is shown in figure 1. The numbers indicate degree of a major scale with 1 representing the root, b3 representing a flat 3rd, etc. In this configuration the notes in a horizontal line form a minor-7 chord. Since adjacent regions for each finger position are musical half steps, changing to a major chord simply means sliding the second finger up to a new region. Below the slider regions is a "mute pad" which can be likened to muting the strings of a guitar. When a palm rests on this area the release time of the sound is shortened in proportion to the pressure applied. Thus, notes can be allowed to decay slowly or muted quickly.

Additional tunings which are changed by selecting one of the four buttons, will be familiar to other string instrument players. The second mapping creates intervals between the sliders in fourths similar to how the four lowest strings of a guitar are tuned, while the third mapping produces intervals of fifths which mirrors the intervals of the violin, viola and cello. The fourth mapping creates intervals of seconds which allows for dense chord structures to be easily played.

Simply pressing fingers down on the desired notes results in sustained notes. The surface is also conducive to swiping across a set of sliders which can be likened to strumming.

4.3 Left Hand

As with most string instruments the left hand is charged with determining the "fret" position. The common string techniques of hammer-ons and pull-offs available. Additionally, because the sensor is continuous, vibrato is easily achieved by wavering the fretting finger back and forth.

In addition to selecting the root note with the left hand, the FSR enables another mode of expression by mapping finger pressure to volume and brightness depending how hard it is pressed. This allows the intensity of a note to be varied after it has been struck.

The interface controls are mounted on a wooden box containing the Beagle Board and Arduino chip used to synthesize the sound. Carefully placed holes obscure the wiring from the sensors. Thus, on the outside all you see is a power cable, a ¼" jack and the user controls.

4.4. SOUND DESIGN

The PD patch is divided into several sections which correspond to the various controllers. This modular approach made the patch easily scalable to the 4 note polyphony it uses. This layout could be easily scaled beyond the current 4 "strings" available. The left hand slider is mapped to a 2 octave range and each of the right hand sliders have a 4 note range with the aforementioned static and continuous regions.

4.5 Synthesis

For the sound synthesis we chose to pursue an electric guitar inspired sound which was still distinctly electronic. Each voice contains three oscillators: the first two are sawtooth waves separated by an octave and very slightly de-tuned comparable to a 12-string guitar which can sound "jangly" given the extra set of strings not being exactly in a 2:1 relationship. A third oscillator is used for FM synthesis to provide new textures at the players discretion. The sound is then filtered through a series of enveloped low-pass filters.

The pressure applied by the left hand to the fretboard maps to the FM synthesis modulation amount and filter cut-off frequencies. With a physical string the timbre changes depending where it is plucked and with how much force. Since the right hand is often occupied whilst playing Quadrofeelia, we decided to give this timbral control to the left hand.

The result is a range of timbres from mellow tones, to sharp and more complex harmonics.

5. OTHER CONSIDERATIONS

Alternative implementations considered involved a motorized wheel for the left hand which would allow scrolling continuously through a small localized range of the circle of fifths with motorized haptic feedback to assist in locking in tune. This was considered too bulky to be involved in the sliding motion of the left hand.

To improve the familiarity of this instrument to the family of pedal and non-pedal lap steel guitarists, an alternative mechanism that more closely mirrors the plucking of a string and strumming would make for an easier transition and provide additional excitation information with say an FSR tab. Maintaining the ability to bend notes may require this type of design to have a set of sensors for bending strings and another for activating notes.

Finally, extending the instrument by adding an additional 6 membrane sensors would allow for more octaves and more varied chord structures and would also match the most common pedal-steel guitar configurations which use 10 strings.

6. SUMMARY

Combining multiple membrane position sensors in a new way has allowed for a new way to bend individual notes and shape chords. Leveraging the power of the BeagleBoard and the Arduino made it possible to create a self sufficient and portable instrument.

Matching the expressivity of the pedal steel guitar in a simple interface that could be played by a beginner helped shape the final product.

7. ACKNOWLEDGMENTS

Special thanks to Edgar Berhdal and Wendy Ju for their help and generosity in sharing their knowledge during the development process, not to mention the CCRMA Satellite platform used in this project.

8. ADDITIONAL AUTHORS

Michael Repper
CCRMA
Stanford University
Stanford, CA
michael.repper@gmail.com

9. REFERENCES

[1] Jakes Bejoy, Kapil Krishnamurthy, and Dan Schlessinger, CCRMA 250a 2008, (https://ccrma.stanford.edu/courses/250a/moviearchive/Aut08/Glide.mov

[2] Kurzweil Music Systems, RBN 1Super Ribbon Programmable Controller (www.kurzweilmusicsystems.com)

[3] Loic Kessous, Julien Castet, Daniel Arfib, "'GXtar', an interface using guitar techniques", Proceedings of the 2009 Conference on New Instruments for Musical Expression, 2009.

SQUEEZY: Extending a Multi-touch Screen with Force Sensing Objects for Controlling Articulatory Synthesis

Johnty Wang
Media and Graphics
Interdisciplinary Centre, UBC
University of British Columba,
Vancouver BC, Canada
johnty@ece.ubc.ca

Nicolas d'Alessandro
Media and Graphics
Interdisciplinary Centre,
University of British Columba,
Vancouver BC, Canada
nda@magic.ubc.ca

Sidney Fels
Media and Graphics
Interdisciplinary Centre,
University of British Columba,
Vancouver BC, Canada
ssfels@ece.ubc.ca

Bob Pritchard
Media and Graphics
Interdisciplinary Centre,
University of British Columba,
Vancouver BC, Canada
bob@interchange.ubc.ca

ABSTRACT
This paper describes Squeezy: a low-cost, tangible input device that adds multi-dimensional input to capacitive multi-touch tablet devices. Force input is implemented through force sensing resistors mounted on a rubber ball, which also provides passive haptic feedback. A microcontroller samples and transmits the measured pressure information. Conductive fabric attached to the finger contact area translates the touch to the bottom of the ball which allows the touchscreen to detect the position and orientation. The addition of a tangible, pressure-sensitive input to a portable multimedia device opens up new possibilities for expressive musical interfaces and Squeezy is used as a controller for real-time gesture controlled voice synthesis research.

Keywords
Musical controllers, tangible interfaces, force sensor, multi-touch, voice synthesis.

1. MOTIVATION
With the increasing popularity of portable multi-touch tablet devices such as the Apple iPad, a large number of developers and researchers are working on mobile applications. The ever-increasing processing power and multimedia capabilities of these devices not only allow more demanding computations, but their convenient form factor and touch input also provide new modes of interaction for the user. Although a multi-touch screen offers a rich set of hand gesture interactions, current implementations lack force input that can be useful for many musical applications. Additionally, interaction with a touch-screen is usually done with virtual widgets inside the screen, and adding a physical object can provide a more intimate user experience [3]. For our particular application, force control provides a suitable input for exploring activation parameters for articulatory speech synthesis.

2. RELATED WORK
The AudioPad [8] is a table-top tangible interface that consists of physical knobs that are tracked using RF resonance. The position and orientation of multiple pucks are tracked and used to control audio synthesis. Another similar interface is the reacTable [4], where physical pucks are tracked visually using cameras. In comparison, Squeezy achieves the tracking using the built-in feature of the touch screen which results in a more portable package.

PreSense [11] combines a capacitive sensor with a resistive sensor and can detect the location and force of a single finger press. A piezo-actuator provides tactile feedback. By using a touchscreen, Squeezy can detect the orientation in addition to position of more than one object.

The SqueezeOrb [9] is a wired force controller built into a hand exerciser. Although it only measures a single force input, the device is complemented with a 6 DOF optical motion tracker.

3. SYSTEM COMPONENTS
The Squeezy, as shown in Figure 1, consists of a modified rubber stress ball augmented with electronics. The stress ball allows the user to squeeze the device and force sensitive resistors (FSR) measure the applied force. The finger pressure exerted on the ball create resistance changes which are connected as a part of a voltage divider and the output voltage is measured by the analog input of an Arduino Pro Mini microcontroller. Passive force feedback is provided through the elastic nature of the ball. The microcontroller is connected to a Bluetooth serial port that wirelessly transmits the measured pressure.

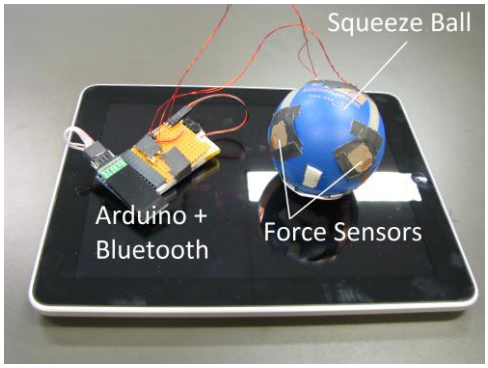


Figure 1. The Squeezy Controller on the iPad

Conductive fabric connected via copper tape towards the bottom of the ball as shown in Figure 2 transfer the finger touches when the user holds Squeezy. Spots of conductive fabric are exposed at the bottom, which is cut flat so it can rest on the touchscreen.

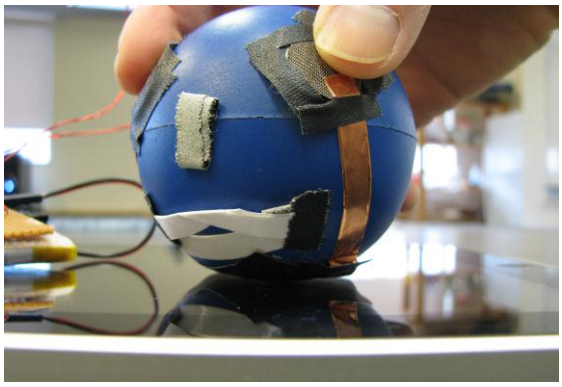


Figure 2. Finger touch transfer mechanism

The position and orientation of Squeezy are determined by the location of the touch spots detected on screen when a user's fingers are placed on the conductive pads. As a first prototype, two points are implemented for each Squeezy ball and hence 180 degrees of rotation can be detected (Using three points would allow detection of the full 360 degrees of rotation). The following table shows the number of points required for the detection of each feature:

Table 1. Number of touch points per Squeezy

# of points	X-Y Position	0-180° orientation	0-360° orientation
1	Yes	No	No
2	Yes	Yes	No
3	Yes	Yes	Yes

The maximum number of detectable touch points (11 on the iPad) and the number of touch points per Squeezy limit the total number Squeezy's that can be used at a time.

The X-Y values of each touch spot is transmitted via Wi-Fi using OSC [7] implemented in a simple openFrameworks [6] application running on the iPad. For the force values, a simple Max/MSP [5] patch parses the Bluetooth serial stream and translates the microcontroller's analog readings. For visual testing of the system, a Processing [2] sketch displays the position and force values.

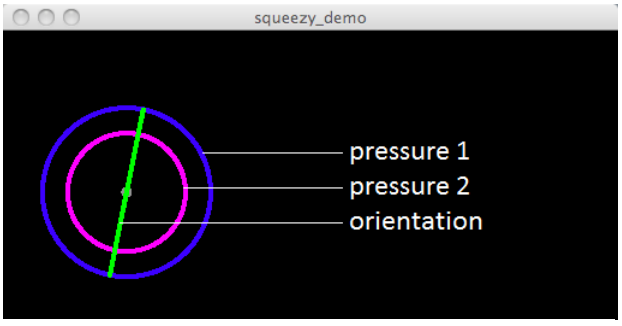


Figure 3. Location, orientation and pressure visualization in a Processing sketch

In Figure 3, the centre of the circles corresponds to the physical position, calculated by the midpoint between the two touch spots. The line across the circle represents the orientation, and the radii of the two circles are proportional to the pressure detected by each force sensor.

In theory, the entire end-application could reside on the tablet device. However, due to the iPad's lack of support for the Bluetooth Serial Port Profile (SPP), the force sensor values had to be sent to another computer first. This would not be necessary for other devices, or an HID Bluetooth device (which the iPad does support).

The Squeezy electronics are powered by an 850mAH lithium polymer battery, and has an estimated runtime of around 8 hours. The total cost of Squeezy is less than $100.

4. APPLICATION

The Squeezy is currently used as an experimental input device to drive a biomechanical model of the vocal tract for speech synthesis as a part of the DiVA project [10]. Currently the 2D position of the Squeezy is mapped to a vowel space while one force sensor drives the pitch. In the next stage of development the squeezing forces will be applied to muscle activations of a bio-mechanical vocal tract model as developed in [1].

5. CONCLUSION

By making use of the physical nature of capacitive touchscreen technology, a simple tangible interface is implemented using low-cost components to complement the input system of a popular tablet device. Concepts similar to Squeezy can be used to extend interfaces to provide more expressive controllers.

6. REFERENCES

[1] Fels, S. et al. Artisynth: Towards Realizing an Extensible, Portable 3D Articulatory Speech Synthesizer. In *International Workshop on Auditory Visual Speech Processing*. July 2005,119-124.

[2] Fry, B., and Reas, C. Processing. http://processing.org

[3] Ishii, H., and Ullmer, B. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, Atlanta, Georgia, 1997, 234-241.

[4] Jordà, S. et al. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international Conference on Tangible and Embedded interaction*, Baton Rouge, Louisiana, February 15-17, 2007.

[5] Max/MSP, Cycling 74. http://cycling74.com/products/maxmspjitter/

[6] Openframeworks Library. http://www.openframeworks.cc

[7] Open Sound Control. http://www.opensoundcontrol.org

[8] Patten, J., Recht, B., and Ishii, H. AudioPad: a tag-based interface for musical performance. In *Proceedings of the 2002 conference on New Interfaces for Musical Expression*, Dublin, Ireland, May 24-26, 2002.

[9] Pintaric, T., Kment, T., and Spreicer W. SqueezeOrb: A Low-Cost Pressure-Sensitive User Input Device. In *Proceedings of the 15th ACM Symposium on Virtual Reality Software and Technology (VRST)*, Bordeaux, France, October 27-29, 2008.

[10] Pritchard, B., and Fels, S. GRASSP: gesturally-realized audio, speech and song performance. In *Proceedings of the 2006 conference on New interfaces for Musical Expression*, Paris, France, June 4-8, 2008.

[11] Rekimoto, J., and Schwesig, C. PreSenseII: Bi-directional Touch and Pressure Sensing Interactions with Tactile Feedback. In *CHI'06 extended abstracts on Human factors in computing systems*, Montréal, Québec, Canada, April 22-27, 2006.

SWAF: Towards a Web Application Framework for Composition and Documentation of Soundscape

Souhwan Choe
Dept. of Digital Contents Convergence
Seoul National University
Seoul, Republic of Korea
virii47@snu.ac.kr

Kyogu Lee
Dept. of Digital Contents Convergence
Seoul National University
Seoul, Republic of Korea
kglee@snu.ac.kr

ABSTRACT
In this paper, we suggest a conceptual model of a Web application framework for the composition and documentation of soundscape and introduce corresponding prototype projects, SeoulSoundMap and SoundScape Composer. We also survey the current Web-based sound projects in terms of soundscape documentation.

Keywords
soundscape, web application framework, sound archive, sound map, soundscape composition, soundscape documentation.

1. INTRODUCTION
Soundscape research was initiated by Schafer [4] during the late 1960s and early 1970s with the aim of enhancing our acoustic environment and human awareness of surrounding sounds. In music, soundscape composition often refers to electroacoustic music which is composed by organizing environmental sounds. Truax [5] described the term soundscape composition as a continuum of "'found sound' representation of acoustic environments through to the incorporation of highly abstracted sonic transformations". In contrast, soundscape as a documentation of sonic environments is a new perspective to artistic expression. Soundscape is not only a record of acoustic phenomena, but also an aggregate of cultural, social, and historical events of a specific place at a specific time. Thus, the practice of documenting sonic environment is a creative activity of narrative writing with sounds.

From the beginning of the Web in 1990s, the Web space has been regarded as one of the important places for artistic practices with its intrinsic capacity of telepresence [1][2]. Online map services and their Open APIs (Application Programming Interface) stimulated the development of interactive sound maps and map-based sound archives on the Web such as UK SoundMap[1], London Sound Survey[2], Open Sound New Orleans [3], SeoulSoundMap [4], and Sons de Barcelona [5]. Urban Tapestries [3] project is an attempt to incorporate elements of social research into creative projects merging the concerns of art and design with social science. SoundTransit[6] is another noticeable Web-based audio project that presents new possibilities of the employment of the Web for the sonic experience by providing the visitors with a sonic journey adopting a metaphor of a flight trip.

2. SWAF (SOUNDSCAPE WEB APPLICATION FRAMEWORK)
SWAF (Soundsacpe Web Application Framework) is a conceptual framework for soundscape composition and documentation. The purpose of this framework is to provide various features such as signal processing, music composition model, ontological sound classification, database management, and user interface for archiving, navigating, composing, and listening to soundscapes.

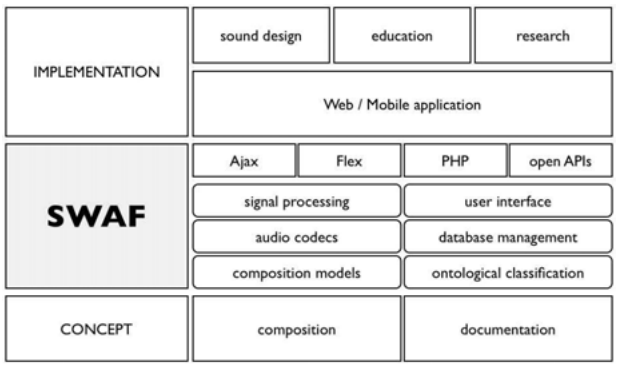
2.1 Overview of SWAF


Figure 1. Overview of SWAF (conceptual model).

The development of SWAF focuses on two aspects: composition and documentation of soundscape as artistic practices. In this respect, the core of SWAF consists of four modules including *Archive*, *Navigator*, *Composer*, and *Community* as illustrated in Figure 2. *Archive* is a module for the management of archived recordings, and thus it usually works in the background to analyze and process the information of each sound and sound itself. In the *Navigator* module the audience can explore soundscapes in various ways, for instance, by a map interface, chronically ordered list, and list by a keyword. *Composer* is similar to a music production application in its functionality; users can compose their own music (or virtual soundscape) by selecting and organizing sound objects from *Navigator* with some sound effects. *Community* is a place where users can share their own compositions and discuss on soundscape and its relevant topics. The prototype of *Navigator* and *Composer* modules have been implemented in the following projects.

[1] http://sounds.bl.uk/uksoundmap
[2] http://www.soundsurvey.org.uk
[3] http://www.opensoundneworleans.com/core
[4] http://som.saii.or.kr/campaign
[5] http://barcelona.freesound.org
[6] http://www.soundtransit.nl

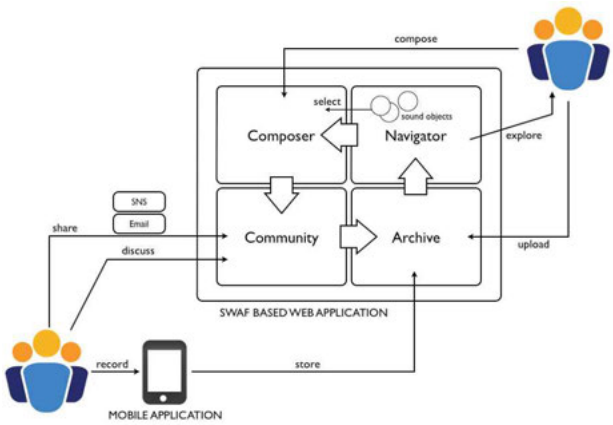


Figure 2. A model of a SWAF-based Web application.

2.2 SeoulSoundMap

SeoulSoundMap is a project to build a collaborative archive of soundscape in Seoul. We utilized several Web technologies for interactive user experience in the development of the project website. Anyone can contribute to the archive and explore the archived soundscapes as well in a geographical or chronical way. If the user wants to share the sonic experience, it is also possible to share any recordings on Twitter.

The website is implemented by mashing up Google Maps, Audioboo[7], and Ajax. Users can collect an instant soundscape in real time using an Audioboo's mobile application, then send the recording with a predefined tag to Audioboo's server. When the webpage is loaded, a PHP script retrieves relevant sounds from Audioboo and shows the retrieved recordings on Google Maps with a marker according to their geographical information. The information window on Google Maps which gives the user some information about the recording is implemented using a JavaScript library jQuery[8].

Figure 3. Screenshot of the SeoulSoundMap webpage.

2.3 SoundScape Composer

SoundScape Composer is an experiment to design a ready-to-use tool for soundscape composition. In designing the SoundScape Composer, we adopted the Schafer's categorization of soundscape into keynote sounds, signals and soundmarks with a slight modification. In this application, sounds on Google Maps represent soundmarks that refer to the community sounds reflecting the unique sonic environment of a specific place in Seoul. As one soundmark is chosen by a spectator, a keynote sound is automatically generated according to the atmosphere of the chosen soundmark. Then users can locate sound objects, refering to Schafer's signal, from the object navigator below the composition palette, and modify each sound object's properties such as position, pitch, volume, and distance to make music.

SoundScape Composer consists of three parts, including the composition palette, the sound map, and the sound object navigator. The graphical user interface is implemented in Flash with Google Maps, and the sound synthesis engine is implemented in SuperCollider. When a user chooses a marker from Google Maps or a sound object from the navigator, an OSC (OpenSoundControl) message is passed to Supercollider, then SuperCollider manages the corresponding sound by controlling Buffer and Synth objects. SuperCollider is also used to generate the keynote sound by means of granular synthesis.

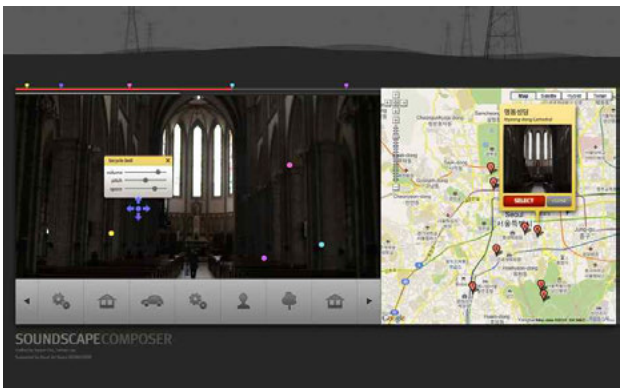


Figure 4. SoundScape Composer (installation version).

3. CONCLUSION

By leveraging the current Web technologies, various web-based sound projects are proposed as a new approach both for sound and music research and artistic expression. In this paper, we suggested a conceptual model of a Web application framework and introduced corresponding prototype projects, SeoulSoundMap and SoundScape Composer. The main goal of SWAF is to provide a robust and extensible framework for the development of Web applications and Web services in terms of soundscape composition and documentation. Future works on SWAF include the implementation of a standard software framework and the development of our own online archive and its applications.

4. ACKNOWLEDGEMENTS

SeoulSoundMap is developed for the Sound@Media project which is an initiative of Moonji Cultural Institute supported by the Seoul Foundation for Arts and Culture. Soundscape Composer is a project commissioned by Seoul Art Space Geumcheon for the exhibition <The Return of Techne>.

5. REFERENCES

[1] Donati, L.P. and Prado, G. Artistic Environments of Telepresence on the World Wide Web. *Leonardo*, 34, 5 (Oct. 2001), 437–442.

[2] Kac, E. Telepresence Art. http://www.ekac.org/telepresence.art._94.html.

[3] Lane, G. Urban Tapestries: Wireless Networking, Public Authoring and Social Knowledge. *Personal and Ubiquitous Computing*, 7, 3-4 (Jul. 2003), 169–175.

[4] Schafer, R.M. *The Soundscape: Our Sonic Environment and the Tuning of the World*. Destiny Books, Rochester, VT, 1993.

[5] Truax, B. Genres and Techniques of Soundscape Composition as Developed at Simon Fraser University. *Organised Sound*, 7, 1 (Apr. 2002), 5–14.

[7] http://audioboo.fm
[8] http://jquery.com

Playing the "MO" – Gestural Control and Re-Embodiment of Recorded Sound and Music

Norbert Schnell, Frederic Bevilacqua, Nicolas Rasamimana,
Julien Blois, Fabrice Guedy, Emmanuel Flety
IRCAM – CNRS STMS
Real Time Musical Interactions
1, place Igor Stavinsky
75004 Paris, France
Norbert.Schnell@ircam.fr

ABSTRACT

We are presenting a set of applications that have been realized with the *MO* modular wireless motion capture device and a set of software components integrated into Max/MSP. These applications, created in the context of artistic projects, music pedagogy, and research, allow for the gestural re-embodiment of recorded sound and music. They demonstrate a large variety of different "playing techniques" in musical performance using wireless motion sensor modules in conjunction with gesture analysis and real-time audio processing components.

Keywords

Music, Gesture, Interface, Wireless Sensors, Gesture Recognition, Audio Processing, Design, Interaction

1. INTRODUCTION

The development of motion capture and sensor technology linking body movements and gestural expression to digital technology has created novel opportunities in the performing arts over the last decades. Since the early experiments by artists like Cage, Cunningham, and Rauschenberg in the 1960s [7] – still using analogue technology – numerous artistic and technological projects have explored novel relationships between movements and sounds using wireless real-time motion capture and interactive sound synthesis. Many of these applications explore bodily involvement on the borderlines between music making and music listening as well as on the intersection of multiple disciplines such as the design and performance of musical instruments, dance, and audiovisual installations.

With the *MO* [8] we propose a device that can be easily customised by a set of accessories. This system allows for the adaptation of the device to a large range of scenarios and applications without requiring strong engineering competences. In this sense, we can present the *MO* as filling the gap between complex development platforms such as the *Arduino* and ready-made gaming controllers such as the *Wiimote*.

The study of the control and embodiment of sound and music by movements in music performance, music listening, and dance as well as extra-musical activities became a vivid field of research over the past years [4]. Research projects have explored applications allowing for the active participation in music listening [1]. The relationship between free gestures and sound have been studied for example from the perspective of *air-playing* or *sound tracing* [5]. Further research has focussed on sound and gestures related to the manipulation of objects [2]. While this research creates the scientific background for many of our applications, the experiments in this domain themselves represent interesting application of the presented technology.

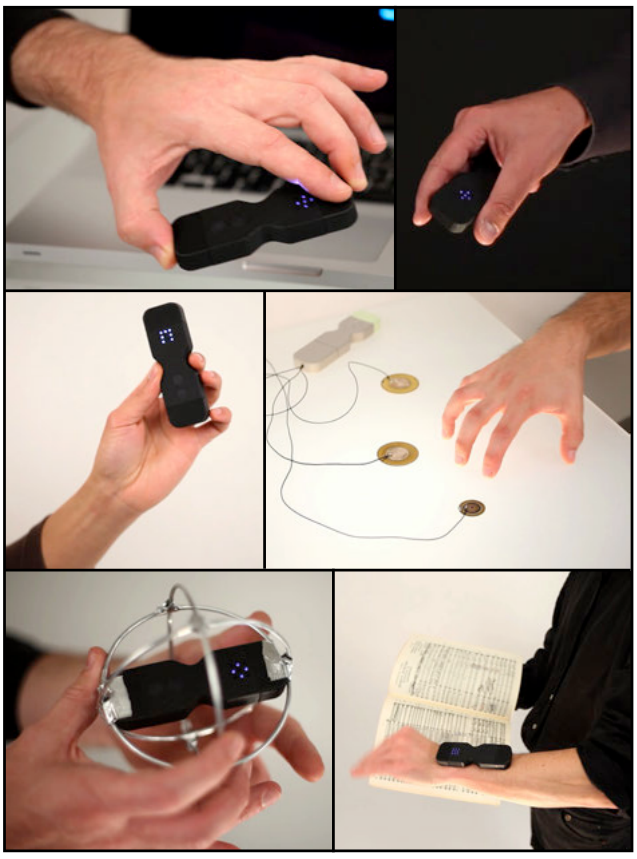


Figure 1: MO modular wireless sensor device used in different playing scenarios.

Many of our developments have been driven by applications in music pedagogy. Methods of music pedagogy such

as the *Eurhythmics* developed by Jaques-Dalcroze [6] can be cited as precursors for many approaches to sound and music embodiment beyond the repertoire of gestures and movements involved in common musical practices (i.e. instrument playing techniques and conducting gestures). Our particular interest in wireless motion capture and interactive audio technology in this context is the possibility to spontaneously create performance scenarios that fit a particular pedagogical goal. In many of our projects, the music students have been involved in the elaboration of collaborative scenarios that focus on particular aspects of music interpretation such as tempo and phrasing as well as more abstract musical aspects such as musical form or harmony. These scenarios often use extra-musical metaphors of playing such as a ball game or a chess match.

Music and audio games are currently emerging as a field of application for many technologies and techniques developed in the domain of digital art. Recently, movement and gesture analysis have become an important aspect in the development of interfaces for gaming platforms and mobile devices. In addition, we observe the appearance of computer games with a strong accent on musical content as well as games dedicated to real-time sound control and music performance. In many aspects, the presented applications are prototyping scenarios, techniques and practices for musical games and playful sound environments for gaming platforms and mobile devices. Other than many platforms, the presented technology supports playing scenarios involving multiple players facing each other in arbitrary spacial setups.

2. APPLICATIONS & SCENARIOS

The presented applications are a summary of different scenarios that we have developed over the past few years. They include musical ball games, virtual orchestra conducting, as well as a wide range of playing techniques based on musical and extra-musical playing metaphors. In these scenarios for one or multiple players, the *MO* wireless sensor module may be hand-held, attached to he body, or used to augment objects such as a ball, a kitchen utility or a table to create new instruments and playing techniques.

3. HARDWARE & SOFTWARE

All presented applications use the *MO* sensor modules with the available accessories including a module with piezo sensors, a baton of LED lights, and a set of passive elements that can be attached to the core module. The presented software components include a set of modules for the analysis and recognition of gestures [3] as well as a set of audio analysis and re-synthesis modules integrated into Max/MSP [9]. The audio processing components developed in the framework of this project are mainly dedicated to the real-time interactive rendering and transformation of recorded sounds. They have been developed for the experimentation with the re-embodiment of recoded sounds and music by gestures and movements. A set of tools for the analysis and annotation of recorded auto content provides automatic extraction of audio descriptors and segmentation as well as graphical visualisation and editing. The real-time modules for the interactive content based rendering and transformation of annotated audio materials include a phase vocoder as well as granular synthesis and concatenative synthesis modules. In addition to the realised applications, the presentation shows how to create new scenarios and playing techniques with the software components developed around the *MO* sensor modules.

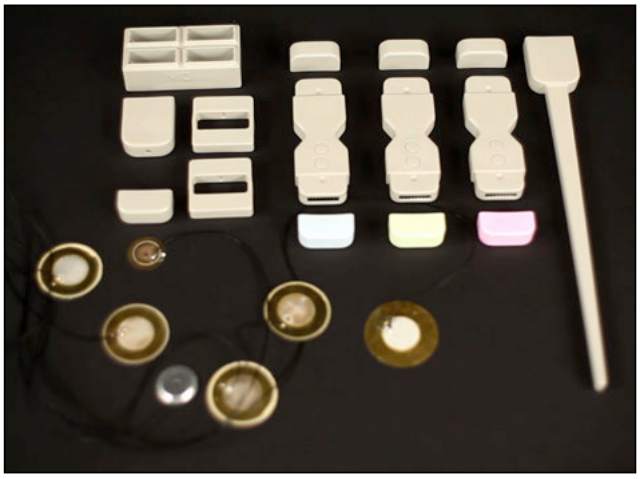


Figure 2: The MO device with different accessories.

4. ACKLOWLEDGEMENTS

The work described in this article has been partially supported by the French National Research Agency (ANR) in the framework of the Interlude project . We would like to thank the involved project partners Ateliers des Feuillantines, GRAME, DaFact, NoDesign, and VoxLer for the inspiring collaboration. Many thanks also to Riccardo Borghesi and the other colleagues of our team.

5. REFERENCES

[1] A. Camurri, C. Canepa, and G. Volpe. Active listening to a virtual orchestra through an expressive gestural interface: The orchestra explorer. In *NIME*, 2007.

[2] F. F. Davide Rocchesso. *The sounding object*. Mondo Estremo, Italy, 2003.

[3] F. Bevilacqua et al. Continuous realtime gesture following and recognition. In *Gesture in Embodied Communication and Human-Computer Interaction (LNCS)*. Springer Verlag, 2009.

[4] R. I. Godoy and M. L. (eds.). *Musical Gestures: Sound, Movement, and Meaning*. Routledge, New York, 2009.

[5] R. I. Godøy, E. Haga, and A. R. Jensenius. *Playing "Air Instruments": Mimicry of Sound-Producing Gestures by Novices and Experts*, volume 3881/2006. Springer-Verlag, Berder Island, France, May 2005.

[6] M.-L. Juntunen. *Embodiment in Dalcroze Eurhythmics*. PhD thesis, University of Oulu, Finland, 2004.

[7] S. Lacerte. *9 Evenings and experiments in Art and Technology : A gap to fill in art history's recent chronicles*. Artists as Inventors/Inventors as Artists. Hatje Cantz/Ludwig Boltzmann Insititut, Berlin/Linz, 2008.

[8] N. Rasamimanana, F. Bevilacqua, N. Schnell, F. Guedy, E. Flety, C. Maestracci, B. Zamborlin, J.-L. Frechin, and U. Petrevski. Modular musical objects towards embodied control of digital music. In *Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, 2011.

[9] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi. Mubu & friends - assembling tools for content based real-time interactive audio processing in max/msp. In *ICMC*, Montreal, Août 2009.

(LAND)MOVES

Bruno Zamborlin *
Goldsmiths/IRCAM
SE146NW London UK
bruno.zamborlin@ircam.fr

Giorgio Partesana †
http://glp.oivil.eu
gioparte@gmail.com

Marco Liuni ‡
IRCAM
1, place Igor Stravinsky - 75004 Paris, France
marco.liuni@ircam.fr

ABSTRACT

(land)moves is an interactive installation: the user's gestures control the multimedia processing with a total synergy between audio and video synthesis and treatment.

Keywords

mapping gesture-audio-video, gesture recognition, landscape, soundscape

1. INTRODUCTION

The project **(land)moves** is an installation where sounds and images evolve according to the same set of control parameters. Such parameters are deduced from the analysis of the user's gestures, and are interpreted in real time by the audio and video engines: human gestures influence the evolving audiovisual landscape. By interacting with a flock of polygons in the video foreground, the user gradually learns to use the device and to influence the evolution of the whole audio-video environment. Gestures, sounds and video create objects with a joint identity: we refer to these entities as a *move*. The user interacts with the multiple media of each move depending on his will and attitudes, but still perceiving a coherent reaction to his movements.

In the next section we describe the artistic foundation of the installation and the user's experience that we aim to exploit. The third section is a summary of the features exchanged between the gestures analysis and recognition engine and the sound and video processing.

2. ARTWORK EXPERIENCE

On the one hand there is the user, on the other an audiovisual landscape: their relation consists in a multimodal interaction. The user is responsible for the landscape modeling through his gestures (see http://www.youtube.com/watch?v=CfKQCAxizrA for the video game *From Dust*, exploiting the idea to model planet Earth). On the other hand, the landscape also conveys a mood (see an example in the opening scene of *Gerry* by Gus Van Santis http://www.youtube.com/watch?v=-_JiB4N-0Ro); therefore, certain features of our landscape are modeled by the user's gestures, which are the expression of an emotional state partially determined by the landscape itself.

2.1 Scenario

The system reproduces a planet that is continuously changing, with a time dimension consisting in cycles of days and nights. The gesture analysis is performed according to two time-scales, which control two distinct but dependent audio-visual levels: the first, constituting the visual foreground, consists of a flock of polygons flying over the landscape; the second background level is the evolving land (see figure 1). The division between the foreground and the background affects the audio domain as well: different sounds and treatments contribute for creating a perception of distance and motion, as detailed in the third section. Some features of user's actions have immediate consequences, while the whole performance of the user is analyzed and affects the whole system in a long-term scale.

This first realization of **(land)moves** is based on two classes of moves: every performed gesture is placed at a certain distance from these classes, which acts as a parameter for the audio and video engines. This classification is performed by the gesture recognition algorithm detailed in section 3.1.

2.2 Space and Interface

The artwork is controlled by a single user, standing in front of a vertical projection of approximately 4 meters width and 2 meters height based on the floor (we are considering the possibility of a multi-users mode for future versions); other visitors can nevertheless access the space. This placement is intended to provide the user for a natural feeling of a landscape. There are no devices to handle, the motion is captured by a Microsoft *Kinect* sensor device.

3. REAL-TIME GESTURE BASED MULTIMEDIA PROCESSING

The implementation consists in three separated applications which communicate through the OSC protocol: the gesture analysis algorithm generates the control messages, which are then sent to the audio and video engines.

3.1 Gestures Analysis

We describe here how the arm movements of the user are represented and translated into control messages. The system takes advantage of the Microsoft Kinect which interprets 3D scene information from a continuously-projected infrared structured light [2]. The main purpose in using this device is to analyze arm gestures without asking to the user to hold a physical device. The PrimeSense OpenNI software framework (http://www.openni.org/) has been used for robustly tracking the hand position providing the first

*Gesture analysis and recognition.

†Visual Artist.

‡Sound Artist.

NIME'11, 30 May–1 June 2011, Oslo, Norway.

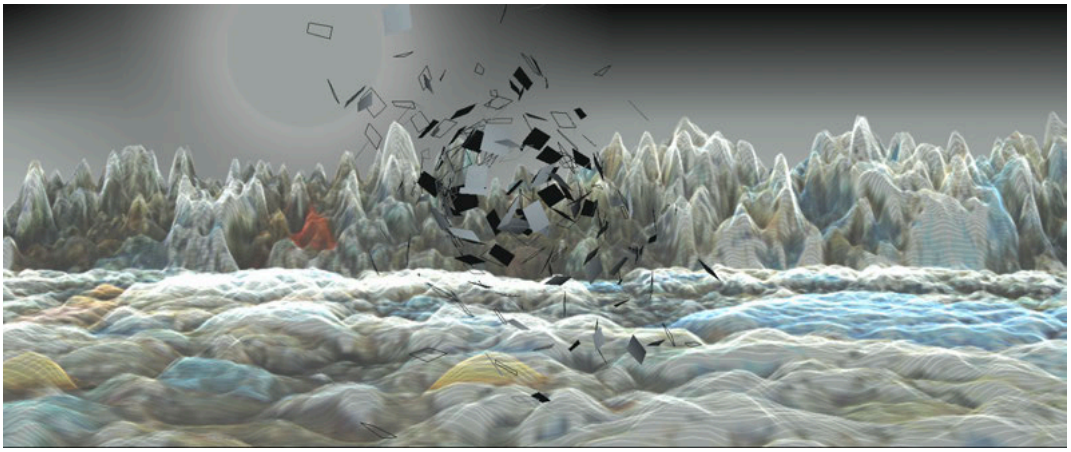


Figure 1: A screenshot of (land)moves .

control signal as 3D position. Based on this information, the quantity of movement of the user is calculated. Two types of quantity of movements are given: *instantaneous* and *global*. The first one is computed over a sliding window of few hundreds of milliseconds and it is used to control parameters that need to react rapidly to user behavior. The second one is computed over a larger scale and gives information related to the global activity of the user's performance. In a similar way, other descriptors based on the symmetry and the velocity of the gesture are calculated. The utilization of these two pieces of information is described in sections 3.2 and 3.3.

Furthermore, the system uses machine learning techniques to estimate a high level quality of movement, called the *roundness/sharpness ratio*. We consider here different gestures, each one assigned to a different value of this movement quality. We use the HMM-based Gesture Follower [1] for continuous likelihood recognition of these gestures.

The final information that is given to the audio and video engines is a high level description of the roundness/sharpness ratio information. This quality reaches here a human meaning, and defines two reference classes of moves: *round* and *sharp* gestures. As the Gesture Follower continuously returns a likelihood estimation referred to each one of these gestures, the final information is continuously interpolated and intermediate values between different signs are possible.

3.2 Video Synthesis

Quartz Composer's Particle System is used to create the foreground objects which are more reactive and engage the user in a direct interaction. The landscape in background, which slowly but constantly mutates, is created by extruding and color correcting photographic textures. This process reproduces some morphologic treats of real world despite the digital polygonal feeling, creating a troubled perception of an imaginary planet. Textures are chosen among an indexed database according to gesture characteristics on a long term analysis.

On the foreground level, gesture features such as velocity and amplitude influence the movement of the flock, making the polygons fly faster or wider following the movements of the arm. Other examples of the mapping for the landscape in background use the energy and the roundness quality defined in section 3.1 of gestures to determine whether the land shapes as gentle hills or spiky mountains. A further element which defines the image is light; the lighting system consists of three elements, the sun/moon, the sky and the ambient light.

3.3 Audio Treatments

The audio feedback is generated from both pre-recorded materials and real time processing: electric guitar samples are used to provide the device for a concrete instrumental feeling. The whole sound engine is driven by gestural descriptors, at different time levels: the sounds in the foreground react to short term descriptors of energy and roundness/sharpness ratio, while those in the background are treated according to features on a longer period. Sounds are classified according to the two classes of moves defined in 3.1. With this choice, we aim to establish two distant points in an appropriate space of timbres, which represent the two classes of gestures, as well as the two possible shapes of the flock of polygons used in the video feedback: all the transitions in between are realized with a source-filter technique based on the *SuperVP* phase vocoder [3].

4. ACKNOWLEDGMENTS

The work described in section 3.1 has been realized in collaboration with Parag Mital from the Goldsmiths EAVI group pkmital.com

Gesture Follower has been developed and is currently maintained by the IMTR team at IRCAM. More information at http://imtr.ircam.fr/imtr/Gesture_Follower

The gesture analysis framework described in 3.1 takes full advantage of FTM. See ftm.ircam.fr for details

SuperVP is developed by Axel Roebel and the Analysis/Synthesis team at IRCAM

Kineme comunity for the development of the particle system http://kineme.net/

Vade for Rutt Etra plugin. v002 Rutt/Etra attempts to emulate the Rutt/Etra raster-based analog synthesizer http://v002.info/?page_id=19

5. REFERENCES

[1] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. Continuous realtime gesture following and recognition. *Gesture in Embodied Communication and Human-Computer Interaction, Springer*, pages 73–84, 2010.

[2] P. MS. Primesense supplies 3-d-sensing technology to project natal for xbox 360. *MsPress*, 2010.

[3] A. Roebel. A shape-invariant phase vocoder for speech transformation. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, September 2010.

Can Haptics make New Music? - Fader and Plank Demos

Bill Verplank
CCRMA - Stanford University
660 Lomita Dr.
Stanford, CA 94305
verplank@ccrma.stanford.edu

Francesco Georg
CCRMA - Stanford University
660 Lomita Dr.
Stanford, CA 94305
fgeorg@stanford.edu

ABSTRACT
Haptic interfaces using active force-feedback have mostly been used for emulating existing instruments and making conventional music. With the right speed, force, precision and software they can also be used to make new sounds and perhaps new music.

The requirements are local microprocessors (for low-latency and high update rates), strategic sensors (for force as well as position), and non-linear dynamics (that make for rich overtones and chaotic music).

Keywords
NIME, Haptics, Music Controllers, Microprocessors.

1. INTRODUCTION
For more than fifteen years, we have been exploring the use of haptics (active force feedback) in music controllers. Recently, an experienced composer spent a morning exploring the latest "Plank". He was surprised: *"the instrument is not only responsive, it's also assertive; and I don't know of another situation like that."* What are the qualities that excited this composer? What was required?

2. EARLY HAPTICS FOR MUSIC
Our most common experience of haptics is in mobile phones - early pagers had vibration for a silent alert. Video games have "rumble packs" for excitement. An inexpensive motor simply spins an eccentric weight. These "tactors" have also been used as simple musical feedback for controlling performance [1]. Beyond vibration, d-c motors can provide constant forces - call it "active force-feedback".

We have attempted to make electronic instruments feel like their physical ancestors. For example, springs and weights were added to electric keyboards - i.e. "passive haptics". Active force-feedback keyboards have been attempted by Cadoz [2] and Gillespie [3]. A four-degree-of-freedom haptic violin was built by Charles Nichols in 2000 [4]. None of these have made it into musical performance.

Two recent examples are notable because with inexpensive components, and modern electronics they achieve surprising results. They are "high-performance haptics".

3. HIGH PERFORMANCE HAPTICS

3.1 Haptic Drum
When you hit Ed Berdahl's haptic drum[5] with a stick, it kicks back using a speaker coil (woofer). Depending on which target you hit with the stick, the drum will make a different sound. By varying the way you hold the stick, the Haptic Drum enables you to play drum rolls that would otherwise be difficult or impossible. For instance, drum rolls can be played at speeds of up to 70Hz. It's superhuman but still a drum.

3.2 Cellomobo
In Collin Oldham's Cellomobo [6], the "bow" is a wooden dowel, the "string" is the blade of a painter's palette knife with a piezo pickup attached. The signal goes via Arduino to Pd where it is delayed by an amount determined by his left hand on a "string" sensor acting as a linear potentiometer. Finally, the delayed signal is amplified and a shaker vibrates the palette knife orthogonal to the motion of the bow, making it stick and slip. This is a (surprising) emulation of an ancient instument.

4 NEW MUSIC

4.1 FM Fader Synth
At Stanford's CCRMA, in Music 250A Wendy Ju and Ed Berdahl attached an Arduino to a BeagleBoard [7] As a student exercise, Francesco Georg programmed a haptic landscape in Pd. With the fader, he "throws" the fader knob as it "bounces" over a landscape of FM synthesis parameters.

When people tried it, there were a variety of surprised expressions: "It's fighting me" or "we're dancing". A German offered the word "widerspenstig" which has something to do with a spirit working against you. A rough translation is "unruly" or "assertive".

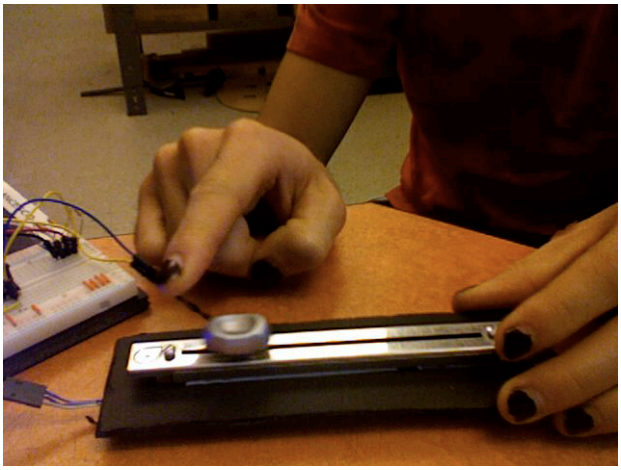


Figure 1. Francesco Georg "throwing" a motorized fader.

4.2 Granular synthesis with the PLANK.

"The PLANK" [8] is made from an old hard-disk drive by removing the disks and using the head-positioning voice-coil actuator to move a cylindrical surface. A force-sensitive resistor senses the force of the user's fingers on the surface. With a simple program on the AVR controller, if you push "into" the surface, it "sides down" a virtual profile - e.g. the envelope of a wave or sample.

In a recent workshop [9] with Bill Verplank and David Zicarelli, Roger Reynolds used the PLANK with Hans Tutschku's granular synthesis (running in MAX/MSP). As Roger bounced the PLANK around the envelope of the sample, he remarked that "it's' a situation in which the instrument is not only responsive, it's also assertive".

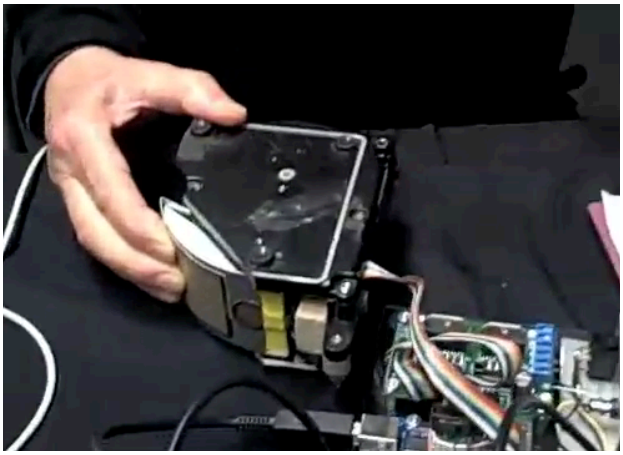


Figure 2. The PLANK with Arduino and Motorboard

Figure 3 shows the display in MaxMSP of the sample which is granularized, the computed envelope and the "slope" smoothed and stored on the Arduino for force feedback.

5. SUMMARY

The combination of inexpensive actuators (motorized fader, disk head-positioner), high-performance hardware, and open software make "high-performance haptics" accessible to musicians. Rather than emulating traditional instruments, we can explore new forms of expressive and lively music.

6. ACKNOWLEDGMENTS

Thanks to Chris Chafe, Max Mathews, Michael Gurevich, Wendy Ju, Ed Berdahl and the students of "Physical Interaction Design for Music".

7. REFERENCES

[1] Birnbaum, D., *Musical vibrotactile feedback*. MA Thesis, McGill, 2007.

[2] Cadoz, C., Lisowski, L., & Florens, J. (1990). "A modular feedback keyboard design". *Computer Music Journal, 14 (2): 47-51.*

[3] Gillespie, B. The Touchback Keyboard, *Internaional* Computer Music Conference, 1992.

[4] Nichols, C. "The vBow: Development of a Virtual Violin Bow Haptic Human-Computer Interface." *NIME-02: Proceedings of the Conference on New Interfaces for Musical Expression*, ACM Press (2002).

[5] Berdahl, E., Steiner, H-C, Oldham, C. Practical Hardware and Algorithms for Creating Haptic Musical Instruments, *Proceedings of the 2008 conference on New interfaces for Musical Expression*. ACM Press (2008).

[6] Oldham, C. Cellomobo. <http://homepage.mac.com/coldham/klang/cellomobo.html

[7] Berdahl, E., Ju, W., *https://ccrma.stanford.edu/~eberdahl/Satellite/*

[8] Verplank, B., Gurevich, M. and Mathews, M. The PLANK: designing a simple haptic controller. *NIME '02: Proceedings of the Conference on New Interfaces for Musical Expression*, ACM Press (2002).

[9] *Incubator 2010: Beyond the instrument metaphor: new paradigms for interactive media.* Arizona State University, February 19-21, 2010. <http://ame.asu.edu/events/incubator/>..

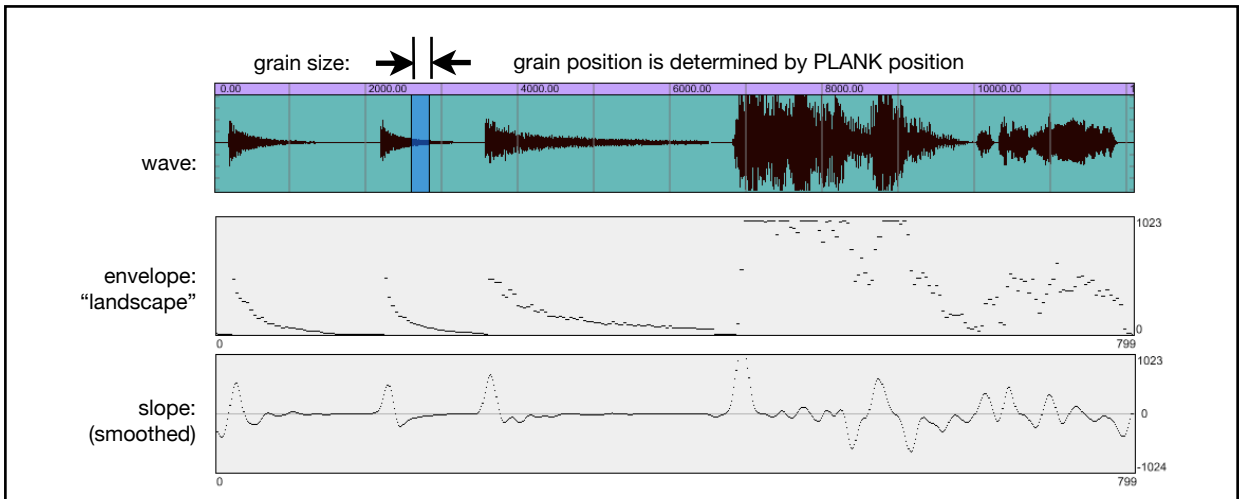


Figure 3. MaxMSP display of "landscape" for granular synthesis with the Plank.

Concerts

Sunday 29 May 18:00

Norwegian Museum of Science, Technology and Medicine

LOLC
Akito van Troyer, Jason Freeman, Avinash Sastry, Sang Won Lee, Shannon Yao

Little Soldier Joe
Øyvind Brandtsegg and Carl Haakon Waadeland

Random Access Solo
Malin Stattin and Gerhard Eckel

Opening of SID exhibition

Reactable
Carles López

Monday 30 May, 17:30–19:00

Oslo City Hall

Cornerghostaxis#1
Gerriet K. Sharma, David Pirrò and Dana Jessen

I-phone improvisation (App: Curtis Lite)
NIME 2011 participants

Monday 30 May, 20:00–23:00

Chateau Neuf

Licht & Hiebe
Jacob Selle and Stefan Weinzierl (Venue: Lillesalen)

ROYGBIV
Joshua Clayton (Venue: Lillesalen)

With Winds (for soprano t-stick)
Andrew Stewart (Venue: Biblioteket)

L'instant
Tom Mays (Venue: Biblioteket)

All Hail the Dawn
Alexander Dupuis (Venue: Lillesalen)

Ural Power
Yoichi Nagashima (Venue: Lillesalen)

Television Sky
EP trio – Erika Donald, Ben Duinker and Eliot Britton (Venue: Biblioteket)

About Place
Michael Straus (Venue: Biblioteket)

Tuesday 31 May, 19:00–20:30

Lindemansalen, Norwegian Academy of Music

Body Jockey
Sarah Taylor, Maurizio Goina and Pietro Polotti

Improvisation for piano + motion capture system
Sarah Nicolls and Nick Gillian

Socks and Ammo
Můstek – Lauren Sarah Hayes and Christos Michalakos

SoundGrasp
Thomas Mitchell and Imogen Heap

TURN ME! I need 12 Volts!
Kristin Norderval

Tuesday 31 May, 21:00–23:00

Chateau Neuf

E=MCH
Paul Stapleton, Caroline Pugh, Adnan Marquez-Borbon and Cavan Fyans (Venue: Lillesalen)

REMI Sings
Christopher Alden (Venue: Biblioteket)

Suspended Beginnings
Diemo Schwarz and Victoria Johnson (Venue: Biblioteket)

The Loop
Jason Dixon, Tom Davis, Jason Geistweidt and Alain B. Renaud (Venue: Klubbscenen)

Dissonance
Victor Zappi and Dario Mazzanti (Venue: Betong)

The Shells
Alex Nowitz (Venue: Biblioteket)

BiLE (Birmingham Laptop Ensemble)
Julien Guillamat, Charles Céleste Hutchins, Shelly Knotts, Norah Lorway, Jorge Garcia Moncada, Chris Tarren (Venue: Lillesalen)

Where Art Thou?: Dance Jockey
Yago de Quay and Ståle Skogstad (Venue: Biblioteket)

Sonolume
Domenico Sciajno (Venue: Lillesalen)

Wednesday 1 June 19:00–20:30

Lindemansalen, Norwegian Academy of Music

Trondheim Voices
Tone Åse, Siri Gjære, Live Maria Roggen, Heidi Skjerve, Ingrid Lode, Kirsti Huke, Anita Kaasbøll, Silje R. Karlsen

Interstices AP
Bill Hsu and Alain Crevoisier

Flayed/Flock
Bill Hsu, Håvard Skaset, Guro Skumsnes Moe

L2Ork
Ivica Ico Bukvic (Director), John Elder, Hillary Guilliams, Bennett Layman, David Mudre, Steven Querry, Philip Seward, Andrew Street, Elizabeth Ullrich and Adam Wirdzek

Wednesday 1 June 21:00–23:00

Chateau Neuf

V'Oct(Ritual)
Mark Bokowiec and Julie Wilson-Bokowiec (Venue: Betong)

mikro:strukt
Satoshi Shiraishi and Alo Allik (Venue: Betong)

Study No. 1 for Overtone Fiddle
Dan Overholt and Lars Grausgaard (Venue: Klubbscenen)

Distributed Composition #1
Doug Van Nort, Pauline Oliveros and Jonas Braasch (Venue: Betong)

7-of-12 dialectologies
Daniel Schorno and Haraldur Karlsson (Venue: Betong)

TweetDreams
Luke Dahl and Carr Wilkerson (Venue: Klubbscenen)

Installations

ROOM#81

Foyer, Chateau Neuf
Nicolas d'Alessandro and Roberto Calderon

ORFI

Foyer, University Library
MusicalFieldsForever – Anders-Petter Andersson, Birgitta Cappelen, Fredrik Olofsson

BM 0.1

3rd floor, University Library
Leo Peschta

Pre-NIME activities

Art.on.Wires Media Festival

The Art.on.Wires Media Festival is a laboratory, hacker space and meeting point for performing artists, creative media professionals and multimedia engineers. Under the main theme live, distributed and networked art we will explore concepts for remote presence and discuss ideas for merging distant realities into a local performance space.

For five days, May 25-29, 2011 we will host workshops, keynote talks, networked performances, interactive installations and a media hacker lab at Chateu Neuf in Oslo and in parallel at live connected locations world-wide. This year's Art.on.Wires Festival is organised in conjunction with the NIME'2011 conference. The festival will end the day before NIME starts on May 30 and NIME participants will be able to attend Art.on.Wires for a reduced fee.

At our main festival site in Betong we have access to a large club and performance space equipped with a 11 x 6.5 m stage and facilities to host an audience of several hundred people. The site is co-located with the Norwegian Music Academy in the town centre of Oslo and can be reached easily by public transport. For live video/audio/data streams from and to the festival site we will provide free Gigabit Internet access.

The festival is organized by the Art.on.Wires Society, Simula Research Laboratory and the University of Oslo, and supported by the Norwegian Research Council.

Symposium: Technology and Aesthetics

Technology and Aesthetics is an international symposium which aims to focus on how advances in technology provide a framework within which music, film and other artistic fields are developing – ushering through changes in the way sonic and visual art is perceived. This symposium takes a closer look at what these changes are – and where we are likely to be headed.

Contributors to the symposium are people who have helped define novel expressions in various artistic fields, or who have spent years commenting on these changes. The roster includes composers, music performers, film sound designers, academics and editors of peer-reviewed journals. The program includes the following speakers: Leigh Landy, Barry Truax, Natasha Barrett, Arnt Maasø, Nicolas Collins, Øyvind Brandtsegg, Cristoph Cox, Randy Thom and Gisle Tveito.

In conjunction with the symposium, there will be evening programs, with a concert featuring Barry Truax (composer), Natasha Barrett (composer) and Leigh Landy (composer and editor-in-chief for Organised Sound) and a screening of Apocalypse Now! with an introduction by Randy Thom (sound designer and sound mixer).

Technology and Aesthetics is arranged as a "pre-NIME event" that leads up to the conference proper. Registration is provided for through our registration system. The symposium is organized by NOTAM – Norwegian Centre for Technology in Music and Arts – in collaboration with the Norwegian Film Institute, and supported by the Norwegian Academy of Music and Arts Council Norway.

Exhibition: Sonic Interaction Design

In connection with NIME 2011, an exhibition on Sonic Interaction Design is curated in collaboration with the EU COST IC0601 Action on Sonic Interaction Design (SID). The exhibition will feature works using sonic interaction within arts, music and design as well as examples of sonification for research and artistic purposes. The exhibition will take place at the Norwegian Museum of Science, Technology and Medicine in Oslo, and will open on 29th May 2011. The call for works was very successful with more than 100 submissions. Twelve works have been selected (pending confirmations of technical and spatial detail).

The Exhibition is curated by Trond Lossius (BEK - Bergen Center for Electronic Arts, Norwegian SID delegate) and Frauke Behrendt (CoDE - The Cultures of the Digital Economy Institute, German SID delegate and chair of Work Group 3). The exhibition is produced by BEK in collaboration with the museum, and is supported by the Norwegian Arts Council and the The COST IC0601 Action on Sonic Interaction Design. The exhibition is also generously supported by part of the COST 'Year of Visibility.'

Tutorials and Workshops

There will be a number of tutorials and workshops in the days leading up to the conference. These are available to both NIME participants and other interested people. Please see below for details about the different workshops.

Hardware Hacking Workshop

A hands-on workshop in "handmade electronic music", tailored for the NIME audience. Assuming no technical background whatsoever, this workshop guides the participants through a series of sound-producing electronic construction projects, including: a) The "Victorian synthesizer" (making an oscillator with just a speaker and a battery). b) "Laying of hands" on a radio circuit board to make the poor man's Cracklebox. c) Basic, versatile oscillator circuit controlled by a wide range of sensors, including potentiometers, photoresistors, homemade pressure sensors, corroded metal, vegetables, electrodes, etc. This leads to discussion of techniques for interfacing sensors to microcontrollers.

Workshop leader: Nicolas Collins

Soft Controller and Synthesizer Workshop

Our workshop will begin with a discussion of soft circuitry including recent research projects completed by FSP. We will demonstrate fabrication techniques for creating soft circuit controllers, including carding, dry felting and machine and hand sewing with conductive and resistive wools, fabrics, threads and methods of attaching to hardware. The group will be lead through building a simple synthesizer circuit followed by building the soft circuit controller to interface with the hardware. Participants should also feel free to bring projects they have already built and would like to potentially make a soft controller for.

Workshop leaders: Lara Grant and Sarah Grant

Designing Mobile Instruments and Performances in urMus

Mobile smart devices have become widely used and are becoming platforms for interactive music performance. In this workshop we teach the process of building new musical instruments on mobile devices using the meta-environment urMus. This allows performers with modest programming and graphical patching knowledge to learn to quickly and with minimal technical distraction, realize their ideas on iPhones, iPads or Android devices. The goal of the workshop is to show the whole process so that participants end having built their own first mobile instrument, including interface look&feel, interaction modeling and sound design and be ready to play it!

Workshop leaders: Georg Essl and Patrick O'Keefe

Optical Motion Capture Technology

This workshop will explain and demonstrate the current state of the art in optical motion capture technology. It will be divided into two three-hour parts. The first part of this workshop will participants familiarize with the Qualisys Oqus motion capture system and demonstrate the steps from starting the system to annotating recordings. The second part will cover how this motion capture system can be used in artistic applications. Examples of musical instruments based on real-time streaming of mocap data in combination with the Xsens Mocap suit will be presented. Participants will also get the opportunity to develop their own mocap instrument.

Workshop leaders: Birgitta Burger, Kristian Nymoen, Arve Voldsund and Ståle A. Skogstad

Integra Live software for performers and composers

Introduce the Integra Live software for performers and composers. The software is the result of a six-year international collaborative project headed by Birmingham Conservatoire, UK, and supported by EU's Culture 2000 program. NOTAM is one of the six research centers that have participated in the project. The software is a work in progress (beta version has been released), and participants will be invited to provide the workshop leaders with feedback from a user perspective.

Workshop leaders: Dag Henning Kalvøy and Henrik Sundt

Introduction course to some alternative electronic instruments

This workshop offers a short tutorial on eponymous instrument & interface design and related instrumental methods and strategies for articulation, proliferation, expression and response shaping (based on STEIM technology) for live music and sound art performance, together with a parallel investigation into their application in the modalities of 3d video and sound diffusion. The goal of the workshop is for participants to gain hands on knowledge and/or deepen their understanding of the above mentioned issues. The provided instruments help to explore some fundamental questions and to address a set of given tasks.

Workshop leaders: Daniel Schorno and Haraldur Karlsson

New Interfaces for Live Looping

This workshop will explore new interfaces for live looping. We will examine various software and hardware devices for looping in live settings. Techniques and methods will be discussed and new interfaces for looping will be put in practice among participants. We will present customized looping

visualization software and a variety of experimental controllers designed for the live looping performer.

Workshop leaders: Simon Morris and Richard Wilderberg

NIME Primer: A Gentle Introduction to Creating New Interfaces for Musical Expression

This workshop provides a general and gentle introduction to the theory and practice of the design of interactive system for music creationand performance. Our intended audience is newcomers to the field who are interested in starting research projects or artistic activity in this area, as well as members of the public with a more generalinterest. Participants will learn key aspects of the theory and practice of musical interface design by studying case studies taken from the first ten years of the NIME conference.

Workshop leaders: Sidney Fels and Michael Lyons

Mapping Everything Else Workshop

The goal of this workshop is to envision musical instruments in a different way. Rather than starting from technical or musical aspects, this workshop provides a chance to investigate what musical mapping means in terms of navigation, exploration and experience through different modalities. Up to 15 participants make metaphorical and tangible models of instruments, which represent the transformation of human interaction to sound in ways which cannot be normally experienced. Other than a profound interest in musical expression, there are no pre-requisites for participation.

Workshop leaders: Georgios Papadakis, Berit Janssen and Jonathan Reus

Basic Training for Group Improvisation

This workshop is about musical tasks, sound exercises and performing games intended to point out the skills desired in collective improvisation. We will explore, with and without instruments, a set of activities concerning awakeness, forbearance, memory, reactivity and risk. These activities can be used in pedagogical or entertainment contexts.

Workshop leader: Luis Alejandro Olarte

NEXUS: Using Ruby on Rails and HTML5 to Distribute Browser Based Interfaces

NEXUS is a project to leverage the power of canvas-based user interface objects and a Ruby on Rails web application to handle distribution of user interfaces, passing interactions via OSC to and from realtime audio/video processing software. In this way, browser based interactions can be utilized for distribution across a variety of static and mobile devices – making worldwide collaborative creative arts a distinct possibility.

Workshop leader: Jesse Allison

Auditory Augmentation of Everyday Objects with Near Real-time Data

Auditory augmentation is a design and development paradigm for the creation of unobtrusive data representation layers that enhance the sonic characteristic of arbitrary physical objects. Its principal idea is to unobtrusively alter the auditory characteristics of object interactions by digital-born data, rather than introducing a completely new soundscape as it is commonly done in sonification environments. After a short introduction into the paradigm, we will conduct a co-design session in which we plan to come up with several alternative augmentation filter designs. These will be implemented just-in-time by one of the workshop leaders.

Workshop leaders: Till Bovermann, René Tünnermann and Thomas Hermann

A Workshop on NIME Education

As NIME has grown over the years, numerous NIME courses have recently sprung up at universities around the world. This workshop will provide a structured forum for NIME educators to share their approaches, experiences and perspectives on teaching NIME curricula. It will be focused on identifying the major challenges that NIME educators face and on developing innovative ideas to address them. The workshop is organized around three themes: materials – the technological platforms employed in teaching a NIME course; methods – the teaching and learning activities that constitute NIME courses; and matters – thetopics and issues that our courses address.

Workshop leaders: Michael Gurevich, Ben Knapp and Sergi Jordà

Musical performance with the Karlax controller

This is a hands-on workshop allowing each participant to explore some of the gestural possibilities of the Karlax controller by playing various synthesis and processing instruments designed for it by composer/performer Tom Mays. Several different approaches will be presented, musically and technically: some of the instruments explored with the Karlax will be "generative" based on synthesis, samples or sound files, while others will be built on real time processing of acoustic input (instrument or voice). Two Karlax instruments will be available so that duos can be possible. The workshop will take place within an immersive 4 channel sound system allowing spatial manipulation with the instrumental gestures.

Workshop leaders: Tom Mays and Rémi Dury

Hyperimprovisation

Live performances and open discussion based on relevant artistic questions: How do we interact musically with augmented instruments and electronic? How do we create meaningful musical content in computer/electronic-based improvisations? Man vs. machine: Who is taking control? For what musical reasons are musicians using controllers, software and general electronic equipment? Musical presentations by: Victoria Johnson, Alex Gunia, and others.

Workshop leaders: Victoria Johnson and Alex Gunia

Audio-graphic Modeling and Interaction Workshop

This workshop focuses on recent advances and future prospects in modelling and rendering audio-graphic scenes. The convergence of the audio and graphic communities is fostered by the increase in computational resources, cognitive studies on cross-modal perception, and industrial needs for realistic audio scenes. Audio-graphic research is spreading in areas such as games, architecture, urbanism, information visualization, or interactive artistic digital media. We will focus on the representation, interaction, rendering, and perception of scenes in which the audio and graphical components are clearly identified and combined (in contrast to standard multimedia video streams). Accepted papers will invited to submit an extended version to a special issue of the Springer

Journal on Multimodal User Interfaces (JMUI).

Workshop leaders: Roland Cahen, Christian Jacquemin, Diemo Schwarz and Hui Ding

Mapping Digital Musical Instruments with libmapper

This workshop will introduce libmapper, an open-source software library for representing input and output signals on a local network, enabling collaborative development of mapping connections between the signals exposed by gestural interfaces and sound synthesis software. The library comes with plugins from various common audio development environments. We will introduce the functionality of the library and demonstrate its usage in several programming languages and environments. We will show examples of simple and complex mapping as well as machine learning-based connections, using a variety of platforms and controllers. We encourage participants to bring along their laptops and musical interfaces.

Workshop leaders: Joseph Malloch, Stephen Sinclair and Marcelo Wanderley

Workshop on Multi-Modal Data Acquisition for Musical Research

We present a workshop on multi-modal measurement and recording of musical performance. This workshop lasts 3 hours, and it will include a presentation of the current technologies and new techniques for data acquisition and synchronization of music and performing arts research experiments using Qualisys (motion capture), BioControl/Infusion (physiological sensors), Arduino (general data acquisition), Motu (audio), and Polhemus (motion sensing) systems among others. We will demonstrate validated techniques developed by our researchers in order to obtain reliable data for the SIEMPRE (Social Interaction using Music PeRformance Experimentation) European project.

Workshop leaders: Javier Jaimovich, Nick Gillian, Miguel Angel Ortiz, Paolo Coletta and Esteban Maestre

Author Index

Keyword Index

N I M E 2 0 1 1 O S L O

11th International Conference on
New Interfaces for Musical Expression
30 May - 1 June 2011, Oslo, Norway

ISSN 2220-4792
ISSN 2220-4806
ISSN 2220-4814